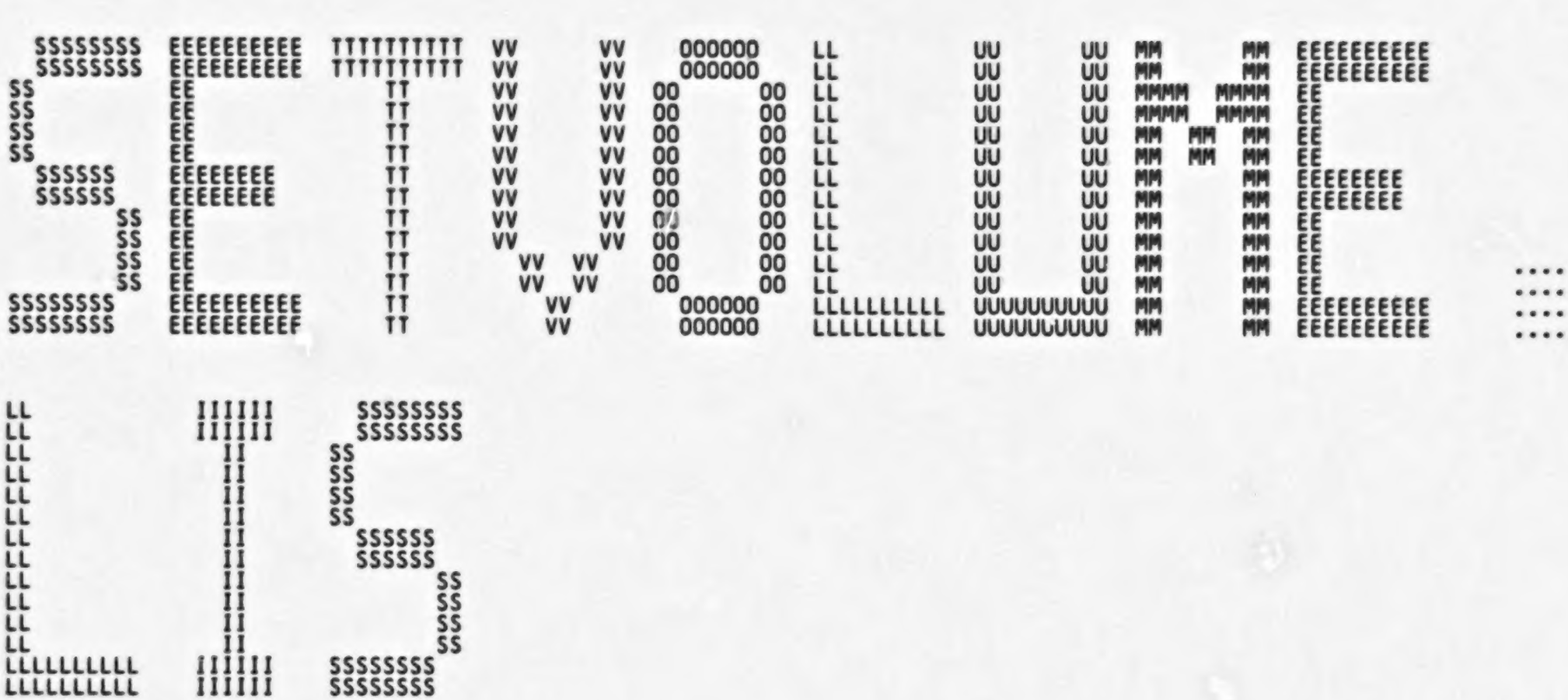
```
   CCCCCCCCCCC  LLL          IIIIIIIII     UUU        UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCC  LLL          IIIIIIIII     UUU        UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCC  LLL          IIIIIIIII     UUU        UUU  TTTTTTTTTTTTTTT  LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
CCC             LLL             III        UUU        UUU        TTT        LLL
   CCCCCCCCCCC  LLLLLLLLLLLLLLL  IIIIIIIII  UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCC  LLLLLLLLLLLLLLL  IIIIIIIII  UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCC  LLLLLLLLLLLLLLL  IIIIIIIII  UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
```


SSSSSSSS EEEEEEEEEE TTTTTTTTTT VV VV 000000 LL UU UU MM MM EEEEEEEEEE
SSSSSSSS EEEEEEEEEE TTTTTTTTTT VV VV 000000 LL UU UU MM MM EEEEEEEEEE
SS EE TT VV VV 00 00 LL UU UU MMMM MMMM EE
SS EE TT VV VV 00 00 LL UU UU MMMM MMMM EE
SS EE TT VV VV 00 00 LL UU UU MM MM MM EE
SSSSSS EEEEEEEE TT VV VV 00 00 LL UU UU MM MM EE
SSSSSS EEEEEEEE TT VV VV 00 00 LL UU UU MM MM EEEEEEEE
SS EE TT VV VV 00 00 LL UU UU MM MM EEEEEEEE
SS EE TT VV VV 00 00 LL UU UU MM MM EE
SS EE TT VV VV 00 00 LL UU UU MM MM EE
SS EE TT VV VV 00 00 LL UU UU MM MM EE ....
SSSSSSSS EEEEEEEEEE TT VV 000000 LLLLLLLLLL UUUUUUUUUU MM MM EEEEEEEEEE ....
SSSSSSSS EEEEEEEEEE TT VV 000000 LLLLLLLLLL UUUUUUUUUU MM MM EEEEEEEEEE ....
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```
;     1          0001  0
;     2          0002  0 MODULE setvol   (
;     3          0003  0                 IDENT = 'V04-000',
;     4          0004  0                 ADDRESSING_MODE(EXTERNAL=GENERAL,
;     5          0005  0                                 NONEXTERNAL=LONG_RELATIVE)
;     6          0006  0                 ) =
;     7          0007  1 BEGIN
;     8          0008  1
;     9          0009  1 !
;    10          0010  1 !*****************************************************************************
;    11          0011  1 !*                                                                           *
;    12          0012  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    13          0013  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    14          0014  1 !*  ALL RIGHTS RESERVED.                                                      *
;    15          0015  1 !*                                                                           *
;    16          0016  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    17          0017  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    18          0018  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    19          0019  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    20          0020  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    21          0021  1 !*  TRANSFERRED.                                                              *
;    22          0022  1 !*                                                                           *
;    23          0023  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    24          0024  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    25          0025  1 !*  CORPORATION.                                                              *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    28          0028  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    29          0029  1 !*                                                                           *
;    30          0030  1 !*                                                                           *
;    31          0031  1 !*****************************************************************************
;    32          0032  1 !
;    33          0033  1
;    34          0034  1 !++
;    35          0035  1 ! FACILITY:     Set Volume Command
;    36          0036  1 !
;    37          0037  1 ! ABSTRACT:
;    38          0038  1 !
;    39          0039  1 !       This module processes the Set Volume command.
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:
;    42          0042  1 !
;    43          0043  1 !       Vax native, privileged user mode
;    44          0044  1 !
;    45          0045  1 !--
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:       Gerry Smith             CREATION DATE: 3-Nov-1981
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 !
;    52          0052  1 !       V03-010 AEW0003         Anne E. Warner          18-Jul-1984
;    53          0053  1 !               Add a check to see if the device specified is a
;    54          0054  1 !               Files-11 format disk and if not tell the user.
;    55          0055  1 !               This check includes the new error message:
;    56          0056  1 !                 set$_notdisk, device is not a files-11 format disk
;    57          0057  1 !
```

```
;    58          0058  1 !
;    59          0059  1 !               Also check to see if qualifiers with 'values' check
;    60          0060  1 !               that the qualifier is present before looking for values.
;    61          0061  1 !               This is because most qualifiers are negatable now.
;    62          0062  1 !               As a result this check was added to /LABEL when it is
;    63          0063  1 !               checked for.
;    64          0064  1 !
;    65          0065  1 !           V03-009 DAS0001         David Solomon           09-Jul-1984
;    66          0066  1 !               Add support for /REBUILD - perform volume rebuild.
;    67          0067  1 !
;    68          0068  1 !           V03-008 AEW0002         Anne E. Warner          24-May-1984
;    69          0069  1 !               Change RMS access to $QIOW access so that the home
;    70          0070  1 !               block can be found in ODS1 structure blocks.  The
;    71          0071  1 !               problem was that RMS sees the End-of-File as zero
;    72          0072  1 !               on an ODS1 initialized volume and will not look for a
;    73          0073  1 !               valid home block.
;    74          0074  1 !
;    75          0075  1 !           V03-007 LMP0221         L. Mark Pilant,         9-Apr-1984  10:46
;    76          0076  1 !               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
;    77          0077  1 !               ORB$W_PROT.
;    78          0078  1 !
;    79          0079  1 !           V03-006 MCN0164         Maria del C. Nasr       03-Apr-1984
;    80          0080  1 !               The /DATA_CHECK qualifier must accept NOREAD and NOWRITE.
;    81          0081  1 !
;    82          0082  1 !           V03-005 AEW0001         Anne E. Warner          21-Mar-1984
;    83          0083  1 !               Add a check to see if volume is mounted foreign.  If
;    84          0084  1 !               it is it cannot be modified because it is not in
;    85          0085  1 !               Files-11 format so notify the user and exit.
;    86          0086  1 !
;    87          0087  1 !           V03-004 GAS0132         Gerry Smith             13-May-1983
;    88          0088  1 !               Add [NO]HIGHWATER, [NO]UNLOAD, [NO]MOUNT_VERIFICATION,
;    89          0089  1 !               [NO]ERASE_ON_DELETE.  Also modify VOLSET.SYS on the
;    90          0090  1 !               root volume for volume sets if /LABEL specified.
;    91          0091  1 !
;    92          0092  1 !           V03-003 GAS0121         Gerry Smith             14-Apr-1983
;    93          0093  1 !               For ODS1 disks, fold long UICs into <377,377>.
;    94          0094  1 !
;    95          0095  1 !           V03-002 GAS0112         Gerry Smith             29-Mar-1983
;    96          0096  1 !               Convert to new CLI interface, and new command dispatcher.
;    97          0097  1 !
;    98          0098  1 !           V03-001 GAS52349        Gerry Smith              4-Jan-1983
;    99          0099  1 !               Remove one level of indirection from the DEVCHAR field
;   100          0100  1 !               of the UCB when modifying its contents.
;   101          0101  1 !
;   102          0102  1 !           V03-006 GAS0091         Gerry Smith             19-Oct-1982
;   103          0103  1 !               Change input request for new CLD syntax.
;   104          0104  1 !
;   105          0105  1 !           V03-005 GAS0040         Gerry Smith              2-Feb-1982
;   106          0106  1 !               Fix privilege checking to check for write access to
;   107          0107  1 !               the volume's index file.  Also, fix write bug that
;   108          0108  1 !               prevented modified home blocks to be written back.
;   109          0109  1 !
;   110          0110  1 !           V03-004 GAS0033         Gerry Smith             12-Jan-1982
;   111          0111  1 !               Fix various bugs.
;   112          0112  1 !
;   113          0113  1 !           V03-003 GAS0030         Gerry Smith              1-Jan-1982
;   114          0114  1 !               Add /RETENTION, the default retention period for files
;                                       created on a volume.
```

```
;  115          0115  1 !
;  116          0116  1 !       V03-002 GAS0026         Gerry Smith             18-Dec-1981
;  117          0117  1 !               Use shared message file, and lower fatal messages to
;  118          0118  1 !               simple error messages.
;  119          0119  1 !
;  120          0120  1 !       V03-001 GAS0025         Gerry Smith             14-Dec-1981
;  121          0121  1 !               Add /LOG qualifier
;  122          0122  1 !
;  123          0123  1 !**
```

```
;  125          0124  1 LIBRARY 'SYS$LIBRARY:LIB';
;  126          0125  1 LIBRARY 'SYS$LIBRARY:TPAMAC';
;  127          0126  1
;  128          0127  1
;  129          0128  1 !
;  130          0129  1 ! ***** Note: The following macro violates the Bliss language definition
;  131          0130  1 ! ***** in that it makes use of the value of SP while building the arg list.
;  132          0131  1 ! ***** It is the opinion of the Bliss maintainers that this usage is safe
;  133          0132  1 ! ***** from planned future optimizations.
;  134          0133  1 !
;  135          0134  1 ! Macro to call the change mode to kernel system service.
;  136          0135  1 ! Macro call format is "KERNEL_CALL (ROUTINE, ARG1, ARG2, ... )".
;  137          0136  1 !
;  138          0137  1 MACRO
;  139        M 0138  1         KERNEL_CALL (R) =
;  140        M 0139  1             BEGIN
;  141        M 0140  1             EXTERNAL ROUTINE SYS$CMKRNL : ADDRESSING_MODE (ABSOLUTE);
;  142        M 0141  1             BUILTIN SP;
;  143        M 0142  1             SYS$CMKRNL (R, .SP, %LENGTH-1
;  144        M 0143  1                         %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI)
;  145          0144  1             END%;
;  146          0145  1
```

```
;  148          0146  1 FORWARD ROUTINE
;  149          0147  1     set$volume : NOVALUE,                   ! Main routine for volume
;  150          0148  1     get_quals,                              ! Get qualifiers
;  151          0149  1     parse_class,                            ! Parse a protection class
;  152          0150  1     process_volume_set : NOVALUE,           ! Process volume set
;  153          0151  1     process_one_volume : NOVALUE,           ! Process each volume
;  154          0152  1     modify_volset : NOVALUE,                ! Fix VOLSET.SYS
;  155          0153  1     set_home,                               ! Modify the homeblock
;  156          0154  1     set_ucbvcb : NOVALUE,                   ! Modify the UCB and VCB for the disk
;  157          0155  1     read_homeblock;                         ! Find and read first good homeblock
;  158          0156  1
;  159          0157  1
;  160          0158  1 EXTERNAL ROUTINE
;  161          0159  1     cli$present,                            ! Get qualifier
;  162          0160  1     cli$get_value,                          ! Get value for qualifier
;  163          0161  1     lib$file_scan,                          ! Routine to get next directory
;  164          0162  1     check_privilege : NOVALUE,              ! Routine to check for privilege
;  165          0163  1     search_error,                           ! Where to go if file search fails
;  166          0164  1     file_error,                             ! Where to go if file error occurs
;  167          0165  1     checksum2,                              ! Compute checksum
;  168          0166  1     get_channelucb,                         ! Routine to get address of UCB
;  169          0167  1     lib$cvt_dtb,                            ! Convert decimal to number
;  170          0168  1     lib$cvt_dtime,                          ! Convert delta time
;  171          0169  1     lib$tparse,                             ! Parser
;  172          0170  1     parse_uic,                              ! Parse a UIC
;  173          0171  1     sys$fao;                                ! Formatted ASCII output
;  174          0172  1
;  175          0173  1 !
;  176          0174  1 ! External data references
;  177          0175  1 !
;  178          0176  1 EXTERNAL
;  179          0177  1 !
;  180          0178  1 ! Data
;  181          0179  1 !
;  182          0180  1     exte_value,                                 ! EXTENSION value
;  183          0181  1     uic_value,                                  ! Owner UIC
;  184          0182  1     group,                                      ! UIC group number
;  185          0183  1     member;                                     ! UIC member number
;  186          0184  1
;  187          0185  1 !
;  188          0186  1 ! Error messages
;  189          0187  1 !
;  190          0188  1 EXTERNAL LITERAL
;  191          0189  1     cli$_ivprot,                        ! Invalid protection value
;  192          0190  1     cli$_absent,
;  193          0191  1     set$_operreq,                       ! OPER privilege required
;  194          0192  1     set$_badfrmt,                       ! Volume doesn't have Files-11 format
;  195          0193  1     set$_hbread,                        ! Error reading homeblock
;  196          0194  1     set$_hbwrite,                       ! Error writing homeblock
;  197          0195  1     set$_modified,                      ! Volume modified
;  198          0196  1     set$_nohome,                        ! Volume has no good home block
;  199          0197  1     set$_notdisk,                       ! Device is not a files-11 format disk
;  200          0198  1     set$_notmod,                        ! Volume not modified
;  201          0199  1     set$_notods2,                       ! Qualifier invalid for ODS1
;  202          0200  1     set$_readerr,                       ! Error reading volume
;  203          0201  1     set$_sysnotupd,                     ! Error updating ucb and vcb
;  204          0202  1     set$_writeerr;                      ! Could not write to file
```

```
;  205          0203  1
;  206          0204  1 !
;  207          0205  1 ! Declare some shared messages
;  208          0206  1 !
;  209        P 0207  1 $SHR_MSGDEF     (SET,119,LOCAL,
;  210        P 0208  1                  (valerr,        error),
;  211        P 0209  1                  (syntax,        error),
;  212        P 0210  1                  (openout,       error),
;  213        P 0211  1                  (closeout,      error),
;  214          0212  1                  (invquaval,     error));
;  215          0213  1
;  216          0214  1
;  217          0215  1 !
;  218          0216  1 ! Literal data definitions
;  219          0217  1 !
;  220          0218  1 LITERAL
;  221          0219  1     true = 1,
;  222          0220  1     false = 0;
;  223          0221  1
;  224          0222  1 LITERAL
;  225        P 0223  1     $EQULST
;  226        P 0224  1          (QUAL_,,1,1,
;  227        P 0225  1          (access,),                  ! ACCESSED bit
;  228        P 0226  1          (data,),                    ! DATA_CHECK bit
;  229        P 0227  1          (exte,),                    ! EXTENSION bit
;  230        P 0228  1          (fprot,),                   ! FILE_PROTECTION bit
;  231        P 0229  1          (label,),                   ! LABEL bit
;  232        P 0230  1          (log,),                     ! LOG bit
;  233        P 0231  1          (owner,),                   ! OWNER_UIC bit
;  234        P 0232  1          (retent,),                  ! RETENTION bit
;  235        P 0233  1          (username,),                ! USER_NAME bit
;  236        P 0234  1          (vprot,),                   ! PROTECTION bit
;  237        P 0235  1          (windows,),                 ! WINDOWS bit
;  238        P 0236  1          (erase,),                   ! [NO]ERASE
;  239        P 0237  1          (erase_val),
;  240        P 0238  1          (fhw,),                     ! [NO]HIGHWATER
;  241        P 0239  1          (fhw_val,),
;  242        P 0240  1          (mntver,),                  ! [NO]MOUNT_VERIFICATION
;  243        P 0241  1          (mntver_val,),
;  244        P 0242  1          (unl,),                     ! [NO]UNLOAD
;  245        P 0243  1          (unl_val,),
;  246        P 0244  1          (rebuild,),                 ! [NO]REBUILD
;  247        P 0245  1          (rebuild_val,),
;  248          0246  1          (lbl_cpy,));                ! Old label was saved
;  249          0247  1 LITERAL
;  250        P 0248  1     $EQULST
;  251        P 0249  1          (DATA_,,1,1,
;  252        P 0250  1          (read,),                        ! DATA_CHECK = READ
;  253        P 0251  1          (write,),                       ! DATA_CHECK = WRITE
;  254        P 0252  1          (noread,),                      ! DATA_CHECK = NOREAD
;  255          0253  1          (nowrite,));                    ! DATA_CHECK = NOWRITE
;  256          0254  1
```

```
;  258          0255  1 !
;  259          0256  1 ! Define storage for this module that must be global
;  260          0257  1 !
;  261          0258  1 GLOBAL
;  262          0259  1     acc_value,                          ! ACCESSED value
;  263          0260  1     fprot_value,                        ! FILE_PROTECTION value
;  264          0261  1     label_value : VECTOR[2],            ! LABEL label
;  265          0262  1     vprot_value,                        ! PROTECTION value
;  266          0263  1     retmin_value : VECTOR[2],           ! Minimum retention period
;  267          0264  1     retmax_value : VECTOR[2],           ! Maximum retention period
;  268          0265  1     user_value : VECTOR[2],             ! USER_NAME
;  269          0266  1     window_value;                       ! WINDOWS
;  270          0267  1
```

```
;  272          0268  1 ;
;  273          0269  1 ! Define own storage for this module
;  274          0270  1 !
;  275          0271  1 OWN
;  276          0272  1     flags : BITVECTOR[32],                  ! Qualifier flags word
;  277          0273  1     dflags : BITVECTOR[32],                 ! DATA_CHECK flags word
;  278          0274  1     user_label : VECTOR[12,BYTE],           ! Place to put username
;  279          0275  1     label_buff : VECTOR[vcb$s_volname,BYTE],    ! Place to store old label
;  280          0276  1     buffer : BLOCK[512,BYTE],               ! Place for home block
;  281          0277  1     acc_inc : BYTE,                         ! Increment to LRU limit
;  282          0278  1     ods1 : BYTE,                            ! ODS1 indicator
;  283          0279  1     channel,                                ! Channel for $QIOW
;  284          0280  1
;  285          0281  1     result_file : VECTOR[nam$c_maxrss,BYTE],
;  286          0282  1
;  287        P 0283  1     NAM : $NAM (RSA = result_file,
;  288          0284  1                 RSS = nam$c_maxrss),
;  289          0285  1
;  290        P 0286  1     FAB : $FAB (DNA = UPLIT BYTE ('[0,0]INDEXF.SYS'),
;  291        P 0287  1                 DNS = %CHARCOUNT ('[0,0]INDEXF.SYS'),
;  292        P 0288  1                 FAC = (get, put, bio),
;  293        P 0289  1                 SHR = (get, upi),
;  294        P 0290  1                 NAM = nam,
;  295          0291  1                 FOP = ufo);
;  296          0292  1
```

```
;   298          0293  1 GLOBAL ROUTINE set$volume : NOVALUE =
;   299          0294  1 !++
;   300          0295  1 !
;   301          0296  1 ! Functional description
;   302          0297  1 !
;   303          0298  1 !       This is the main control module for SET VOLUME.  It obtains the
;   304          0299  1 !       qualifiers and, for each volume specification, calls the routine
;   305          0300  1 !       that actually modifies the volume's home block.
;   306          0301  1 !
;   307          0302  1 ! Calling sequence
;   308          0303  1 !
;   309          0304  1 !       CALL set$volume()
;   310          0305  1 !
;   311          0306  1 ! Input parameters
;   312          0307  1 !       none
;   313          0308  1 !
;   314          0309  1 ! Output parameters
;   315          0310  1 !       none
;   316          0311  1 !
;   317          0312  1 ! Implicit outputs
;   318          0313  1 !       none
;   319          0314  1 !
;   320          0315  1 ! Routine value
;   321          0316  1 !       none
;   322          0317  1 !
;   323          0318  1 ! Side effects
;   324          0319  1 !       none
;   325          0320  1 !
;   326          0321  1 !--
;   327          0322  2 BEGIN
;   328          0323  2
;   329          0324  2 LOCAL
;   330          0325  2     dyn_desc : $BBLOCK[dsc$c_s_bln];
;   331          0326  2
;   332          0327  2 !
;   333          0328  2 ! Check that the image is running with appropriate privilege.
;   334          0329  2 !
;   335          0330  2 check_privilege();
;   336          0331  2
;   337          0332  2 !
;   338          0333  2 ! Get the command qualifiers.
;   339          0334  2 !
;   340          0335  2 IF NOT get_quals()
;   341          0336  2 THEN RETURN;
;   342          0337  2
;   343          0338  2
;   344          0339  2 !
;   345          0340  2 ! For each volume specified, perform the operations requested.
;   346          0341  2 !
;   347          0342  2 $init_dyndesc(dyn_desc);                        ! Make desc. dynamic
;   348          0343  2 WHILE cli$get_value(%ASCID 'VOLUME', dyn_desc)  ! For each volume specified,
;   349          0344  2 DO
;   350          0345  3     BEGIN
;   351          0346  3     LOCAL
;   352          0347  3         status,
;   353          0348  3         max_rvn : volatile,                     ! Total volumes in set
;   354          0349  3         original_rvn : volatile,                ! Original rvn (this disk)
```

```
;  355          0350  1          root_desc : VECTOR[2],                          ! Root volume descriptor
;  356          0351  1          root_buffer : VECTOR[128,BYTE],                 ! Place to put root name
;  357          0352  1          iosb : VECTOR[4,WORD],                          ! Status block for GETDVI
;  358          0353  1          devchar : $BBLOCK [DIB$K_LENGTH],               ! Longword of device characterisitcs
;  359          0354  1                                                          ! defined in $DEVDEF
;  360          0355  1          dvi_list : $ITMLST_DECL(ITEMS=4);               ! $GETDVI item list
;  361          0356  1
;  362          0357  1      !
;  363          0358  1      ! Get the root volume, the total number of volumes, and the volume number of
;  364          0359  1      ! the original volume.
;  365          0360  1      !
;  366        P 0361  1          $ITMLST_INIT(ITMLST = dvi_list,                 ! Set up DVI list
;  367        P 0362  1                       (ITMCOD = dvi$_rootdevnam,         ! Want root volume
;  368        P 0363  1                        BUFADR = root_buffer,             ! name,
;  369        P 0364  1                        BUFSIZ = %ALLOCATION(root_buffer),
;  370        P 0365  1                        RETLEN = root_desc),
;  371        P 0366  1                       (ITMCOD = dvi$_volnumber,          ! this disk's volume
;  372        P 0367  1                        BUFADR = original_rvn),           ! number, and
;  373        P 0368  1                       (ITMCOD = dvi$_volcount,           ! the total number of
;  374        P 0369  1                        BUFADR = max_rvn),                ! volumes.
;  375        P 0370  1                       (ITMCOD = dvi$_devchar,            ! Get the device characteristics
;  376          0371  1                        BUFADR = devchar));               ! to find if mounted foreign.
;  377          0372  1
;  378          0373  1          root_desc[1] = root_buffer;                     ! Set up parameter for
;  379          0374  1                                                          ! later processing.
;  380        P 0375  1          status = $GETDVIW(ITMLST = dvi_list,            ! Get the information.
;  381        P 0376  1                            DEVNAM = dyn_desc,
;  382          0377  1                            IOSB   = iosb);
;  383          0378  1          IF .status
;  384          0379  1          THEN status = .iosb[0];
;  385          0380  1          IF NOT .status                                  ! If a problem, signal,
;  386          0381  1          THEN SIGNAL(.status)                            ! otherwise
;  387          0382  1          ELSE
;  388          0383  1      !
;  389          0384  1      ! If the device specified is not a Files-11 volume or the volume was mounted
;  390          0385  1      ! foreign it cannot be modified, so signal an error and exit.
;  391          0386  1      !
;  392          0387  1
;  393          0388  1      ! Check if a Files-11 volume was specified
;  394          0389  1      !
;  395          0390  1 4        BEGIN
;  396          0391  1 4          IF NOT .devchar[dev$v_rnd]
;  397          0392  1 4          THEN
;  398          0393  1 5          BEGIN
;  399          0394  1 5          LOCAL
;  400          0395  1 5            nodisk_desc : $BBLOCK[dsc$c_s_bln];       ! descriptor for device
;  401          0396  1 5
;  402          0397  1 5            $INIT_DYNDESC (nodisk_desc);
;  403          0398  1 5            nodisk_desc[dsc$w_length] = .root_desc[0];! length of device name
;  404          0399  1 5            nodisk_desc[dsc$a_pointer] = .root_desc[1];! device name
;  405          0400  1 5            SIGNAL                                    ! inform user of error
;  406          0401  1 5             (set$_notmod, 1, nodisk_desc, set$_notdisk);
;  407          0402  1 5            RETURN false;
;  408          0403  1 4          END;
;  409          0404  1 4      !
;  410          0405  1 4      ! It is a Files-11 device so check if mounted foreign
;  411          0406  1 4      !
```

```
;  412          0407  4          IF .devchar[dev$v_for]
;  413          0408  4          THEN
;  414          0409  5          BEGIN
;  415          0410  5          LOCAL
;  416          0411  5            foreign_desc : $BBLOCK[dsc$c_s_bln];          ! descriptor for volume name
;  417          0412  5
;  418          0413  5            $INIT_DYNDESC (foreign_desc);
;  419          0414  5            foreign_desc[dsc$w_length] = .root_desc[0];   ! length of volume name
;  420          0415  5            foreign_desc[dsc$a_pointer] = .root_desc[1];  ! volume name
;  421          0416  5            SIGNAL                                        ! inform user of error
;  422          0417  5              (set$_notmod, 1, foreign_desc, set$_badfrmt);
;  423          0418  5            RETURN false;
;  424          0419  4          END;
;  425          0420  4 !
;  426          0421  4 !  If everything is alright process the volume set.
;  427          0422  4 !
;  428          0423  4          process_volume_set(root_desc
;  429          0424  4                            .original_rvn,
;  430          0425  4                            .max_rvn);
;  431          0426  3        END;
;  432          0427  2      END;
;  433          0428  2
;  434          0429  2 RETURN;
;  435          0430  1 END;


                                                                .TITLE  SETVOL
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

53 59 53 2E 46 58 45 44 4E 49 5D 30 2C 30 5B  00000 P.AAA:    .ASCII  \[0,0]INDEXF.SYS\                       ;
                                              0000F           .BLKB   1
                        00 00 45 4D 55 4C 4F 56  00010 P.AAC:    .ASCII  \VOLUME\<0><0>                          ;
                                        010E0006  00018 P.AAB:    .LONG   17694726                                ;
                                        00000000' 0001C           .ADDRESS P.AAC                                  ;

                                                                .PSECT  $OWN$,NOEXE,2

                                                  00000 FLAGS:    .BLKB   4
                                                  00004 DFLAGS:   .BLKB   4
                                                  00008 USER_LABEL:
                                                                .BLKB   12
                                                  00014 LABEL_BUFF:
                                                                .BLKB   12
                                                  00020 BUFFER:   .BLKB   512
                                                  00220 ACC_INC:  .BLKB   1
                                                  00221 ODST:     .BLKB   1
                                                  00222           .BLKB   2
                                                  00224 CHANNEL:  .BLKB   4
                                                  00228 RESULT_FILE:
                                                                .BLKB   255
                                                  00327           .BLKB   1
                                               02 00328 NAM:      .BYTE   2                                       ;
                                               60 00329           .BYTE   96                                      ;
                                               FF 0032A           .BYTE   -1                                      ;
                                               00 00328           .BYTE   0                                       ;
```

```
00000000' 0032C          .ADDRESS RESULT_FILE
       00  00330          .BYTE   0
       00  00331          .BYTE   0
       00  00332          .BYTE   0
       00  00333          .BYTE   0
00000000   00334          .LONG   0
00000000   00338          .LONG   0
     0000# 0033C          .WORD   0[8]
     0000# 0034C          .WORD   0[3]
     0000# 00352          .WORD   0[3]
00000000   00358          .LONG   0
00000000   0035C          .LONG   0
       00  00360          .BYTE   0
       00  00361          .BYTE   0
       00  00362          .BYTE   0
       00  00363          .BYTE   0
       00  00364          .BYTE   0
       00  00365          .BYTE   0
       00# 00366          .BYTE   0[2]
00000000   00368          .LONG   0
00000000   0036C          .LONG   0
00000000   00370          .LONG   0
00000000   00374          .LONG   0
00000000   00378          .LONG   0
00000000   0037C          .LONG   0
00000000# 00380          .LONG   0[2]
       03  00388 FAB:     .BYTE   3
       50  00389          .BYTE   80
     0000  0038A          .WORD   0
00020000   0038C          .LONG   131072
00000000   00390          .LONG   0
00000000   00394          .LONG   0
00000000   00398          .LONG   0
     0000  0039C          .WORD   0
       23  0039E          .BYTE   35
       42  0039F          .BYTE   66
00000000   003A0          .LONG   0
       00  003A4          .BYTE   0
       00  003A5          .BYTE   0
       00  003A6          .BYTE   0
       02  003A7          .BYTE   2
00000000   003A8          .LONG   0
00000000   003AC          .LONG   0
00000000' 003B0          .ADDRESS NAM
00000000   003B4          .LONG   0
00000000' 003B8          .ADDRESS P.AAA
       00  003BC          .BYTE   0
       0F  003BD          .BYTE   15
     0000  003BE          .WORD   0
00000000   003C0          .LONG   0
     0000  003C4          .WORD   0
       00  003C6          .BYTE   0
       00  003C7          .BYTE   0
00000000   003C8          .LONG   0
00000000   003CC          .LONG   0
     0000  003D0          .WORD   0
       00  003D2          .BYTE   0
```

```
                               00 003D3          .BYTE    0                                           ;
                         00000000 003D4          .LONG    0                                           ;

                                                 .PSECT   $GLOBAL$,NOEXE,2

                                  00000 ACC_VALUE::
                                                 .BLKB    4
                                  00004 FPROT_VALUE::
                                                 .BLKB    4
                                  00008 LABEL_VALUE::
                                                 .BLKB    8
                                  00010 VPROT_VALUE::
                                                 .BLKB    4
                                  00014 RETMIN_VALUE::
                                                 .BLKB    8
                                  0001C RETMAX_VALUE::
                                                 .BLKB    8
                                  00024 USER_VALUE::
                                                 .BLKB    8
                                  0002C WINDOW_VALUE::
                                                 .BLKB    4

                                                 .EXTRN   CLI$PRESENT, CLI$GET_VALUE
                                                 .EXTRN   LIB$FILE_SCAN, CHECK_PRIVILEGE
                                                 .EXTRN   SEARCH_ERROR, FILE_ERROR
                                                 .EXTRN   CHECKSUM2, GET_CHANNELUCB
                                                 .EXTRN   LIB$CVT_DTB, LIB$CVT_DTIME
                                                 .EXTRN   LIB$TPARSE, PARSE_UIC
                                                 .EXTRN   SYS$FAO, EXTE_VALUE
                                                 .EXTRN   UIC_VALUE, GROUP
                                                 .EXTRN   MEMBER, CLI$_IVPROT
                                                 .EXTRN   CLI$_ABSENT, SET$_OPERREQ
                                                 .EXTRN   SET$_BADFRMT, SET$_HBREAD
                                                 .EXTRN   SET$_HBWRITE, SET$_MODIFIED
                                                 .EXTRN   SET$_NOHOME, SET$_NOTDISK
                                                 .EXTRN   SET$_NOTMOD, SET$_NOTODS2
                                                 .EXTRN   SET$_READERR, SET$_SYSNOTUPD
                                                 .EXTRN   SET$_WRITEERR, SYS$GETDVIW

                                                 .PSECT   $CODE$,NOWRT,2

                             0004 00000          .ENTRY   SET$VOLUME, Save R2                         ; 0293
            52 00000000G  00   9E 00002          MOVAB    LIB$SIGNAL, R2                              ;
            5E      FEB0  CE   9E 00009          MOVAB    -336(SP), SP                                ;
00000000G   00            00   FB 0000E          CALLS    #0, CHECK_PRIVILEGE                         ; 0330
00000000V   EF            00   FB 00015          CALLS    #0, GET_QUALS                               ; 0335
            1B            50   E9 0001C          BLBC     R0, 2$                                      ;
      F8    AD 020E0000   8F   D0 0001F          MOVL     #34471936, DYN_DESC                         ; 0342
                    FC    AD   D4 00027          CLRL     DYN_DESC+4                                  ;
                    F8    AD   9F 0002A 1$:      PUSHAB   DYN_DESC                                    ; 0343
               00000000'  EF   9F 0002D          PUSHAB   P.AAB                                       ;
00000000G   00            02   FB 00033          CALLS    #2, CLI$GET_VALUE                           ;
            01            50   E8 0003A 2$:      BLBS     R0, 3$                                      ;
                               04 0003D          RET                                                  ;
            50        08  AE   9E 0003E 3$:      MOVAB    DVI_LIST, $$ITMBLKPTR                       ; 0371
            80 00320080   8F   D0 00042          MOVL     #3276928, ($$ITMBLKPTR)+                    ;
            80      FF68  CD   9E 00049          MOVAB    ROOT_BUFFER, ($$ITMBLKPTR)+                 ;
```

```
                        80        E8   AD  9E 0004E          MOVAB   ROOT_DESC, ($$ITMBLKPTR)+                    ;
                        80 002E0004    8F  D0 00052          MOVL    #3014660, ($$ITMBLKPTR)+                     ;
                        80        F0   AD  9E 00059          MOVAB   ORIGINAL_RVN, ($$ITMBLKPTR)+                 ;
                                       80  D4 0005D          CLRL    ($$ITMBLKPTR)+                               ;
                        80 00300004    8F  D0 0005F          MOVL    #3145732, ($$ITMBLKPTR)+                     ;
                        80        F4   AD  9E 00066          MOVAB   MAX_RVN, ($$ITMBLKPTR)+                      ;
                                       80  D4 0006A          CLRL    ($$ITMBLKPTR)+                               ;
                        80 00020004    8F  D0 0006C          MOVL    #131076, ($$ITMBLKPTR)+                      ;
                        80        3C   AE  9E 00073          MOVAB   DEVCHAR, ($$ITMBLKPTR)+                      ;
                                       80  7C 00077          CLRQ    ($$ITMBLKPTR)+                               ;
                    EC  AD        FF68   CD  9E 00079          MOVAB   ROOT_BUFFER, ROOT_DESC+4                     ; 0373
                                       7E  7C 0007F          CLRQ    -(SP)                                        ; 0377
                                       7E  D4 00081          CLRL    -(SP)                                        ;
                                 FF60  CD  9F 00083          PUSHAB  IOSB                                         ;
                                   18  AE  9F 00087          PUSHAB  DVI_LIST                                     ;
                                   F8  AD  9F 0008A          PUSHAB  DYN_DESC                                     ;
                                       7E  7C 0008D          CLRQ    -(SP)                                        ;
              00000000G            00    08  FB 0008F          CALLS   #8, SYS$GETDVIW                              ;
                                   08    50  E9 00096          BLBC    STATUS, 4$                                   ; 0378
                                   50  FF60  CD  3C 00099          MOVZWL  IOSB, STATUS                         ; 0379
                                   07    50  E8 0009E          BLBS    STATUS, 6$                                   ; 0380
                                         50  DD 000A1 4$:       PUSHL   STATUS                                       ; 0381
                                   62    01  FB 000A3          CALLS   #1, LIB$SIGNAL                               ;
                                         82  11 000A6 5$:       BRB     1$                                           ;
                  1B        3F   AE    04  E0 000A8 6$:       BBS     #4, DEVCHAR+3, 7$                            ; 0391
                                   6E 020E0000  8F  D0 000AD          MOVL    #34471936, NODISK_DESC               ; 0397
                                         04  AE  D4 000B4          CLRL    NODISK_DESC+4                                ;
                                   6E        E8  AD  B0 000B7          MOVW    ROOT_DESC, NODISK_DESC               ; 0398
                                   04  AE    EC  AD  D0 000BB          MOVL    ROOT_DESC+4, NODISK_DESC+4           ; 0399
                                      00000000G  8F  DD 000C0          PUSHL   #SET$_NOTDISK                        ; 0401
                                         1D  11 000C6          BRB     8$                                           ;
                                   28        3F   AE  E9 000C8 7$:       BLBC    DEVCHAR+3, 9$                    ; 0407
                                   6E 020E0000  8F  D0 000CC          MOVL    #34471936, FOREIGN_DESC              ; 0413
                                         04  AE  D4 000D3          CLRL    FOREIGN_DESC+4                               ;
                                   6E        E8  AD  B0 000D6          MOVW    ROOT_DESC, FOREIGN_DESC              ; 0414
                                04 AE    EC  AD  D0 000DA          MOVL    ROOT_DESC+4, FOREIGN_DESC+4          ; 0415
                                      00000000G  8F  DD 000DF          PUSHL   #SET$_BADFRMT                        ; 0417
                                         04  AE  9F 000E5 8$:       PUSHAB  FOREIGN_DESC                             ;
                                         01  DD 000E8          PUSHL   #1                                           ;
                                      00000000G  8F  DD 000EA          PUSHL   #SET$_NOTMOD                         ;
                                   62            04  FB 000F0          CALLS   #4, LIB$SIGNAL                       ;
                                                 04  00F3          RET                                          ; 0418
                                         F4  AD  DD 000F4 9$:       PUSHL   MAX_RVN                              ; 0425
                                         F0  AD  DD 000F7          PUSHL   ORIGINAL_RVN                         ; 0424
                                         E8  AD  9F 000FA          PUSHAB  ROOT_DESC                            ; 0423
                          00000000V      EF    03  FB 000FD          CALLS   #3, PROCESS_VOLUME_SET               ;
                                         A0  11 00104          BRB     5$                                           ; 0343
                                             04  00106          RET                                          ; 0430

; Routine Size:  263 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  437     0431  1 ROUTINE get_quals =
;  438     0432  1 !++
;  439     0433  1 !
;  440     0434  1 ! This routine interrogates the CLI to get all the qualifiers and
;  441     0435  1 ! values.
;  442     0436  1 !
;  443     0437  1 !
;  444     0438  2 BEGIN
;  445     0439  2
;  446     0440  2 BUILTIN
;  447     0441  2     addm,
;  448     0442  2     cmpm;
;  449     0443  2
;  450     0444  2 LOCAL
;  451     0445  2     status,
;  452     0446  2     desc : $BBLOCK[dsc$c_s_bln];
;  453     0447  2
;  454     0448  2 $init_dyndesc(desc);                            ! Make the desc. dynamic
;  455     0449  2
;  456     0450  2 !
;  457     0451  2 ! /ACCESSED
;  458     0452  2 !
;  459     0453  2 IF cli$present(%ASCID 'ACCESSED')
;  460     0454  2 THEN
;  461     0455  2     BEGIN
;  462     0456  2     LOCAL privs : $BBLOCK[8];           ! Place to store the process privileges
;  463     0457  2     flags[qual_access] = 1;
;  464     0458  2
;  465     0459  3 !
;  466     0460  3 ! Call $SETPRV to get the current privileges of the process.  If the process
;  467     0461  3 ! does not have OPER, then signal an error and stop.
;  468     0462  3 !
;  469   P 0463  4     IF NOT (status = $SETPRV(ENBFLG = 1,                ! Enable
;  470   P 0464  4                              PRVADR = 0,                ! No new privileges
;  471   P 0465  4                              PRMFLG = 1,                ! Get current privileges
;  472     0466  4                              PRVPRV = privs))
;  473     0467  3     THEN
;  474     0468  4         BEGIN
;  475     0469  4         SIGNAL(.status);
;  476     0470  4         RETURN false;
;  477     0471  3         END;
;  478     0472  3     IF NOT .privs[prv$v_oper]
;  479     0473  3     THEN
;  480     0474  4         BEGIN
;  481     0475  4         SIGNAL(set$_operreq);
;  482     0476  4         RETURN false;
;  483     0477  3         END;
;  484     0478  3
;  485     0479  3 !
;  486     0480  3 ! The process has the correct privilege, so go ahead and get the value
;  487     0481  3 !
;  488     0482  3     acc_value = 3;                          ! Set up the default
;  489     0483  3
;  490     0484  3 !
;  491     0485  3 ! If a value was specified, use it; otherwise, use the default.
;  492     0486  3 !
;  493     0487  3     IF cli$get_value(%ASCID 'ACCESSED', desc)
```

```
;  494     0488  3      THEN
;  495     0489  4          BEGIN
;  496     0490  4          IF NOT LIB$CVT_DTB(.desc[dsc$w_length],
;  497     0491  4                             .desc[dsc$a_pointer],
;  498     0492  4                             acc_value)
;  499     0493  4          THEN
;  500     0494  5              BEGIN
;  501     0495  5              SIGNAL(set$_syntax, 1, desc);
;  502     0496  5              RETURN false;
;  503     0497  4              END;
;  504     0498  4          IF .acc_value LSS 0                    ! Check that value is in range
;  505     0499  4          OR .acc_value GTR 255
;  506     0500  4          THEN
;  507     0501  5              BEGIN
;  508     0502  5              SIGNAL(set$_syntax, 1, desc, set$_valerr);
;  509     0503  5              RETURN false;
;  510     0504  4              END;
;  511     0505  3          END;
;  512     0506  2      END;
;  513     0507  2
;  514     0508  2  !
;  515     0509  2  ! /DATA_CHECK
;  516     0510  2  !
;  517     0511  2  IF cli$present(%ASCID 'DATA_CHECK')
;  518     0512  2  THEN
;  519     0513  3      BEGIN
;  520     0514  3      flags[qual_data] = 1;
;  521     0515  3      IF NOT cli$get_value(%ASCID 'DATA_CHECK', desc)
;  522     0516  3      THEN
;  523     0517  3          dflags[data_write] = 1
;  524     0518  3      ELSE
;  525     0519  3          WHILE cli$get_value(%ASCID 'DATA_CHECK', desc) DO
;  526     0520  4          BEGIN
;  527     0521  4          IF CH$EQL(.desc[dsc$w_length], .desc[dsc$a_pointer],
;  528     0522  4                    .desc[dsc$w_length], UPLIT(BYTE('WRITE')))
;  529     0523  4          THEN dflags[data_write] = 1
;  530     0524  4          ELSE IF CH$EQL(.desc[dsc$w_length], .desc[dsc$a_pointer],
;  531     0525  4                         .desc[dsc$w_length], UPLIT(BYTE('READ')))
;  532     0526  4          THEN dflags[data_read] = 1
;  533     0527  4          ELSE IF CH$EQL(.desc[dsc$w_length], .desc[dsc$a_pointer],
;  534     0528  4                         .desc[dsc$w_length], UPLIT(BYTE('NOWRITE')))
;  535     0529  4          THEN dflags[data_nowrite] = 1
;  536     0530  4          ELSE IF CH$EQL(.desc[dsc$w_length], .desc[dsc$a_pointer],
;  537     0531  4                         .desc[dsc$w_length], UPLIT(BYTE('NOREAD')))
;  538     0532  4          THEN dflags[data_noread] = 1
;  539     0533  4          ELSE
;  540     0534  5              BEGIN
;  541     0535  5              SIGNAL(set$_syntax, 1, desc);
;  542     0536  5              RETURN false;
;  543     0537  4              END;
;  544     0538  3          END;
;  545     0539  2      END;
;  546     0540  2
;  547     0541  2  !
;  548     0542  2  ! /[NO]ERASE_ON_DELETE
;  549     0543  2  !
;  550     0544  2  status = cli$present(%ASCID 'ERASE_ON_DELETE');
```

```
;  551          0545  2 IF .status NEQ cli$_absent
;  552          0546  2 THEN
;  553          0547  3     BEGIN
;  554          0548  3     flags[qual_erase] = 1;
;  555          0549  3     flags[qual_erase_val] = .status;
;  556          0550  2     END;
;  557          0551  2
;  558          0552  2 !
;  559          0553  2 ! /EXTENSION
;  560          0554  2 !
;  561          0555  2 IF cli$present(%ASCID 'EXTENSION')
;  562          0556  2 THEN
;  563          0557  3     BEGIN
;  564          0558  3     flags[qual_exte] = 1;
;  565          0559  3     exte_value = 5;
;  566          0560  3     IF cli$get_value(%ASCID 'EXTENSION', desc)
;  567          0561  3     THEN
;  568          0562  4         BEGIN
;  569          0563  4         IF NOT lib$cvt_dtb(.desc[dsc$w_length],
;  570          0564  4                            .desc[dsc$a_pointer],
;  571          0565  4                            exte_value)
;  572          0566  4         THEN
;  573          0567  5             BEGIN
;  574          0568  5             SIGNAL(set$_syntax, 1, desc);
;  575          0569  5             RETURN false;
;  576          0570  4             END;
;  577          0571  4         IF .exte_value LSS 0
;  578          0572  4         OR .exte_value GTR 65535
;  579          0573  4         THEN
;  580          0574  5             BEGIN
;  581          0575  5             SIGNAL(set$_syntax, 1, desc, set$_valerr);
;  582          0576  5             RETURN false;
;  583          0577  5             END;
;  584          0578  4         END;
;  585          0579  3     END;
;  586          0580  2
;  587          0581  2 !
;  588          0582  2 ! /FILE_PROTECTION
;  589          0583  2 !
;  590          0584  2 IF cli$present(%ASCID 'FILE_PROTECTION')
;  591          0585  2 THEN
;  592          0586  3     BEGIN
;  593          0587  3     BIND
;  594          0588  3         setpro_mask = fprot_value + 2 : WORD,
;  595          0589  3         setpro_prot = fprot_value : WORD;
;  596          0590  3
;  597          0591  3     flags[qual_fprot] = 1;
;  598          0592  3     fprot_value = 0;
;  599          0593  3
;  600          0594  3     IF cli$present(%ASCID 'FILE_PROTECTION.SYSTEM')
;  601          0595  3     THEN
;  602          0596  4         BEGIN
;  603          0597  4         setpro_mask = .setpro_mask OR %X'000F';
;  604          0598  4         IF cli$get_value(%ASCID 'FILE_PROTECTION.SYSTEM',desc)
;  605          0599  4         THEN setpro_prot = parse_class(desc);
;  606          0600  4         END;
;  607          0601  3     IF cli$present(%ASCID 'FILE_PROTECTION.OWNER')
```

```
;  608          0602  3      THEN
;  609          0603  4          BEGIN
;  610          0604  4          setpro_mask = .setpro_mask OR %X'00F0';
;  611          0605  4          IF cli$get_value(%ASCID 'FILE_PROTECTION.OWNER',desc)
;  612          0606  4          THEN setpro_prot = .setpro_prot OR parse_class(desc)^4;
;  613          0607  3          END;
;  614          0608  3      IF cli$present(%ASCID 'FILE_PROTECTION.GROUP')
;  615          0609  3      THEN
;  616          0610  4          BEGIN
;  617          0611  4          setpro_mask = .setpro_mask OR %X'0F00';
;  618          0612  4          IF cli$get_value(%ASCID 'FILE_PROTECTION.GROUP',desc)
;  619          0613  4          THEN setpro_prot = .setpro_prot OR parse_class(desc)^8;
;  620          0614  3          END;
;  621          0615  3      IF cli$present(%ASCID 'FILE_PROTECTION.WORLD')
;  622          0616  3      THEN
;  623          0617  4          BEGIN
;  624          0618  4          setpro_mask = .setpro_mask OR %X'F000';
;  625          0619  4          IF cli$get_value(%ASCID 'FILE_PROTECTION.WORLD',desc)
;  626          0620  4          THEN setpro_prot = .setpro_prot OR parse_class(desc)^12;
;  627          0621  3          END;
;  628          0622  2      END;
;  629          0623  2
;  630          0624  2  !
;  631          0625  2  ! /[NO]HIGHWATER_MARKING
;  632          0626  2  !
;  633          0627  2  status = cli$present(%ASCID 'HIGHWATER_MARKING');
;  634          0628  2  IF .status NEQ cli$_absent
;  635          0629  2  THEN
;  636          0630  3      BEGIN
;  637          0631  3      flags[qual_fhw] = 1;
;  638          0632  3      flags[qual_fhw_val] = NOT .status;
;  639          0633  2      END;
;  640          0634  2
;  641          0635  2  !
;  642          0636  2  ! /LABEL
;  643          0637  2  !
;  644          0638  2  IF cli$present(%ASCID 'LABEL')
;  645          0639  2  THEN
;  646          0640  2      IF cli$get_value(%ASCID 'LABEL', desc)
;  647          0641  2      THEN
;  648          0642  3      BEGIN
;  649          0643  3      flags[qual_label] = 1;
;  650          0644  3      IF .desc[dsc$w_length] GTR vcb$s_volname
;  651          0645  3      THEN
;  652          0646  4          BEGIN
;  653          0647  4          SIGNAL(set$_syntax, 1, desc);
;  654          0648  4          RETURN false;
;  655          0649  3          END;
;  656          0650  3      label_value[0] = .desc[dsc$w_length];
;  657          0651  3      label_value[1] = .desc[dsc$a_pointer];
;  658          0652  3      $init_dyndesc(desc);
;  659          0653  2      END;
;  660          0654  2
;  661          0655  2  !
;  662          0656  2  ! /LOG
;  663          0657  2  !
;  664          0658  2  flags[qual_log] = cli$present(%ASCID 'LOG');
```

```
;  665     0659  2
;  666     0660  2 !
;  667     0661  2 ! /[NO]MOUNT_VERIFICATION
;  668     0662  2 !
;  669     0663  2 status = cli$present(%ASCID 'MOUNT_VERIFICATION');
;  670     0664  2 IF .status NEQ cli$_absent
;  671     0665  2 THEN
;  672     0666  3     BEGIN
;  673     0667  3     flags[qual_mntver] = 1;
;  674     0668  3     flags[qual_mntver_val] = .status;
;  675     0669  2     END;
;  676     0670  2
;  677     0671  2 !
;  678     0672  2 ! /OWNER_UIC
;  679     0673  2 !
;  680     0674  2 IF cli$present(%ASCID 'OWNER_UIC')
;  681     0675  2 THEN
;  682     0676  3     BEGIN
;  683     0677  3     flags[qual_owner] = 1;
;  684     0678  3     IF NOT cli$get_value(%ASCID 'OWNER_UIC', desc)
;  685     0679  3     THEN
;  686     0680  4         BEGIN
;  687     0681  4         LOCAL
;  688     0682  4             iosb : VECTOR[4,WORD];
;  689   P 0683  4         status = $GETJPIW(ITMLST = UPLIT(WORD(4,jpi$_uic),
;  690   P 0684  4                                          uic_value,
;  691   P 0685  4                                          0,
;  692   P 0686  4                                          0),
;  693     0687  4                          IOSB = iosb);
;  694     0688  4         IF .status
;  695     0689  4         THEN status = .iosb[0];
;  696     0690  4         IF NOT .status
;  697     0691  4         THEN
;  698     0692  5             BEGIN
;  699     0693  5             SIGNAL(.status);
;  700     0694  5             RETURN false;
;  701     0695  4             END;
;  702     0696  4         END
;  703     0697  3     ELSE parse_uic(desc, uic_value);
;  704     0698  2     END;
;  705     0699  2
;  706     0700  2 !
;  707     0701  2 ! /PROTECTION
;  708     0702  2 !
;  709     0703  2 IF cli$present(%ASCID 'PROTECTION')
;  710     0704  2 THEN
;  711     0705  3     BEGIN
;  712     0706  3     BIND
;  713     0707  3         setpro_mask = vprot_value * 2 : WORD,
;  714     0708  3         setpro_prot = vprot_value : WORD;
;  715     0709  3
;  716     0710  3     flags[qual_vprot] = 1;
;  717     0711  3     vprot_value = 0;
;  718     0712  3
;  719     0713  3     IF cli$present(%ASCID 'PROTECTION.SYSTEM')
;  720     0714  3     THEN
;  721     0715  4         BEGIN
```

```
;   722     0716  4          setpro_mask = .setpro_mask OR %X'000F';
;   723     0717  4          IF cli$get_value(%ASCID 'PROTECTION.SYSTEM',desc)
;   724     0718  4          THEN setpro_prot = parse_class(desc);
;   725     0719  3          END;
;   726     0720  3      IF cli$present(%ASCID 'PROTECTION.OWNER')
;   727     0721  3      THEN
;   728     0722  4          BEGIN
;   729     0723  4          setpro_mask = .setpro_mask OR %X'00F0';
;   730     0724  4          IF cli$get_value(%ASCID 'PROTECTION.OWNER',desc)
;   731     0725  4          THEN setpro_prot = .setpro_prot OR parse_class(desc)^4;
;   732     0726  3          END;
;   733     0727  3      IF cli$present(%ASCID 'PROTECTION.GROUP')
;   734     0728  3      THEN
;   735     0729  4          BEGIN
;   736     0730  4          setpro_mask = .setpro_mask OR %X'0F00';
;   737     0731  4          IF cli$get_value(%ASCID 'PROTECTION.GROUP',desc)
;   738     0732  4          THEN setpro_prot = .setpro_prot OR parse_class(desc)^8;
;   739     0733  3          END;
;   740     0734  3      IF cli$present(%ASCID 'PROTECTION.WORLD')
;   741     0735  3      THEN
;   742     0736  4          BEGIN
;   743     0737  4          setpro_mask = .setpro_mask OR %X'F000';
;   744     0738  4          IF cli$get_value(%ASCID 'PROTECTION.WORLD',desc)
;   745     0739  4          THEN setpro_prot = .setpro_prot OR parse_class(desc)^12;
;   746     0740  3          END;
;   747     0741  2      END;
;   748     0742  2
;   749     0743  2 !
;   750     0744  2 ! /[NO]REBUILD
;   751     0745  2 !
;   752     0746  2 status = cli$present(%ASCID 'REBUILD');
;   753     0747  2 IF .status NEQ cli$_absent
;   754     0748  2 THEN
;   755     0749  3     BEGIN
;   756     0750  3     flags[qual_rebuild] = 1;
;   757     0751  3     flags[qual_rebuild_val] = .status;
;   758     0752  2     END;
;   759     0753  2
;   760     0754  2 !
;   761     0755  2 ! /RETENTION
;   762     0756  2 !
;   763     0757  2 IF cli$present(%ASCID 'RETENTION')
;   764     0758  2 THEN
;   765     0759  3     BEGIN
;   766     0760  3     LOCAL temp_desc : VECTOR[2];
;   767     0761  3
;   768     0762  3     flags[qual_retent] = 1;
;   769     0763  3
;   770     0764  3     CH$FILL(0, 8, retmin_value);                ! Zero minimum value
;   771     0765  3     CH$FILL(0, 8, retmax_value);                ! Zero maximum value
;   772     0766  3
;   773     0767  3 !
;   774     0768  3 ! If a minimum value was not supplied, signal an error
;   775     0769  3 !
;   776     0770  3     IF NOT cli$get_value(%ASCID 'RETENTION', desc)
;   777     0771  3     THEN
;   778     0772  4         BEGIN
```

```
;  779     0773  4          SIGNAL(set$_syntax, 1, desc);
;  780     0774  4          RETURN false;
;  781     0775  3          END;
;  782     0776  3
;  783     0777  3      !
;  784     0778  3      ! Convert the minimum retention value to 64-bit system delta time format
;  785     0779  3      !
;  786     0780  3        IF NOT (status = LIB$CVT_DTIME(desc, temp_desc))
;  787     0781  3        THEN
;  788     0782  4            BEGIN
;  789     0783  4            SIGNAL(set$_syntax, 1, retmin_value);
;  790     0784  4            RETURN false;
;  791     0785  4            END
;  792     0786  3        ELSE CH$MOVE(8, temp_desc, retmin_value);   ! If no error, put 64-bit
;  793     0787  3                                                    ! delta time in place
;  794     0788  3
;  795     0789  3      !
;  796     0790  3      ! If a maximum value was supplied, then convert it in the same way.
;  797     0791  3      !
;  798     0792  3        IF cli$get_value(%ASCID 'RETENTION', desc)
;  799     0793  3        THEN
;  800     0794  4            BEGIN
;  801     0795  5            IF NOT (status = LIB$CVT_DTIME(desc, temp_desc))
;  802     0796  4            THEN
;  803     0797  5                BEGIN
;  804     0798  5                SIGNAL(set$_syntax, 1, retmax_value);
;  805     0799  5                RETURN false;
;  806     0800  5                END
;  807     0801  4            ELSE CH$MOVE(8, temp_desc, retmax_value);
;  808     0802  4            END
;  809     0803  4
;  810     0804  4      !
;  811     0805  4      ! If no maximum value was supplied, then use the lesser of:
;  812     0806  4      !      twice the minimum value or
;  813     0807  4      !      the minimum value plus one week
;  814     0808  4      !
;  815     0809  4      !
;  816     0810  3        ELSE
;  817     0811  4            BEGIN
;  818     0812  4            LOCAL
;  819     0813  4                double : VECTOR[2],                         ! Place for 2*RETMIN
;  820     0814  4                week_plus : VECTOR[2],                      ! Place for RETMIN + 7
;  821     0815  4                one_week : VECTOR[2]
;  822     0816  4                    INITIAL(%X'D71BC000',                   ! Binary for 7 days
;  823     0817  4                            %X'FFFFFA7F');
;  824     0818  4            ADDM(2, retmin_value, retmin_value, double);    ! Get 2*RETMIN
;  825     0819  4            ADDM(2, one_week, retmin_value, week_plus);     ! and RETMIN+7
;  826     0820  4            IF CMPM(2, double, week_plus) GTR 0             ! compare...
;  827     0821  4            THEN CH$MOVE(8, double, retmax_value)
;  828     0822  4            ELSE CH$MOVE(8, week_plus, retmax_value);
;  829     0823  3            END;
;  830     0824  2        END;
;  831     0825  2
;  832     0826  2      !
;  833     0827  2      ! /[NO]UNLOAD
;  834     0828  2      !
;  835     0829  2      status = cli$present(%ASCID 'UNLOAD');
```

```
;  836     0830  2 IF .status NEQ cli$_absent
;  837     0831  2 THEN
;  838     0832  2     BEGIN
;  839     0833  2     flags[qual_unl] = 1;
;  840     0834  2     flags[qual_unl_val] = .status;
;  841     0835  2     END;
;  842     0836  2
;  843     0837  2 !
;  844     0838  2 ! /USER_NAME
;  845     0839  2 !
;  846     0840  2 IF cli$present(%ASCID 'USER_NAME')
;  847     0841  2 THEN
;  848     0842  3     BEGIN
;  849     0843  3     flags[qual_username] = 1;
;  850     0844  3     IF NOT cli$get_value(%ASCID 'USER_NAME', desc)
;  851     0845  3     THEN
;  852     0846  4         BEGIN
;  853     0847  4         LOCAL
;  854     0848  4             jpi_list : $ITMLST_DECL (ITEMS = 1),
;  855     0849  4             iosb : VECTOR[4,WORD];
;  856   P 0850  4         $ITMLST_INIT(ITMLST = jpi_list,
;  857   P 0851  4                      (ITMCOD = jpi$_username,
;  858   P 0852  4                       BUFADR = user_label,
;  859   P 0853  4                       BUFSIZ = hm2$s_ownername,
;  860     0854  4                       RETLEN = user_value[0]));
;  861   P 0855  4         status = $GETJPIW(ITMLST = jpi_list,
;  862     0856  4                           IOSB   = iosb);
;  863     0857  4         IF .status
;  864     0858  4         THEN status = .iosb[0];
;  865     0859  4         IF NOT .status
;  866     0860  4         THEN
;  867     0861  5             BEGIN
;  868     0862  5             SIGNAL(.status);
;  869     0863  5             RETURN false;
;  870     0864  4             END;
;  871     0865  4         user_value[1] = user_label;
;  872     0866  4         END
;  873     0867  3     ELSE
;  874     0868  4         BEGIN
;  875     0869  4         IF .desc[dsc$w_length] GTR hm2$s_ownername
;  876     0870  4         THEN
;  877     0871  5             BEGIN
;  878     0872  5             SIGNAL(set$_syntax, 1, desc);
;  879     0873  5             RETURN false;
;  880     0874  4             END;
;  881     0875  4         user_value[0] = .desc[dsc$w_length];
;  882     0876  4         user_value[1] = .desc[dsc$a_pointer];
;  883     0877  4         $init_dyndesc(desc);
;  884     0878  3         END;
;  885     0879  2     END;
;  886     0880  2
;  887     0881  2 !
;  888     0882  2 ! /WINDOWS
;  889     0883  2 !
;  890     0884  2 IF cli$present(%ASCID 'WINDOWS')
;  891     0885  2 THEN
;  892     0886  3     BEGIN
```

```
;  893          0887  3      flags[qual_windows] = 1;
;  894          0888  3      window_value = 7;
;  895          0889  3      IF cli$get_value(%ASCID 'WINDOWS', desc)
;  896          0890  3      THEN
;  897          0891  4          BEGIN
;  898          0892  4          IF NOT lib$cvt_dtb(.desc[dsc$w_length],
;  899          0893  4                             .desc[dsc$a_pointer],
;  900          0894  4                             window_value)
;  901          0895  4          THEN
;  902          0896  5              BEGIN
;  903          0897  5              SIGNAL(set$_syntax, 1, desc);
;  904          0898  5              RETURN false;
;  905          0899  4              END;
;  906          0900  4          IF .window_value LSS 7
;  907          0901  4          OR .window_value GTR 80
;  908          0902  4          THEN
;  909          0903  5              BEGIN
;  910          0904  5              SIGNAL(set$_syntax, 1, desc, set$_valerr);
;  911          0905  5              RETURN false;
;  912          0906  4              END;
;  913          0907  3          END;
;  914          0908  2      END;
;  915          0909  2
;  916          0910  2 RETURN true;
;  917          0911  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                44  45  53  53  45  43  43  41  00020 P.AAE:   .ASCII  \ACCESSED\                    ;
                                                    010E0008    00028 P.AAD:   .LONG   17694728                     ;
                                                    00000000'   0002C          .ADDRESS P.AAE                      ;
                                44  45  53  53  45  43  43  41  00030 P.AAG:   .ASCII  \ACCESSED\                    ;
                                                    010E0008    00038 P.AAF:   .LONG   17694728                     ;
                                                    00000000'   0003C          .ADDRESS P.AAG                      ;
            00  00  4B  43  45  48  43  5F  41  54  41  44      00040 P.AAI:   .ASCII  \DATA_CHECK\<0><0>            ;
                                                    010E000A    0004C P.AAH:   .LONG   17694730                     ;
                                                    00000000'   00050          .ADDRESS P.AAI                      ;
            00  00  4B  43  45  48  43  5F  41  54  41  44      00054 P.AAK:   .ASCII  \DATA_CHECK\<0><0>            ;
                                                    010E000A    00060 P.AAJ:   .LONG   17694730                     ;
                                                    00000000'   00064          .ADDRESS P.AAK                      ;
            00  00  4B  43  45  48  43  5F  41  54  41  44      00068 P.AAM:   .ASCII  \DATA_CHECK\<0><0>            ;
                                                    010E000A    00074 P.AAL:   .LONG   17694730                     ;
                                                    00000000'   00078          .ADDRESS P.AAM                      ;
                                            45  54  49  52  57  0007C P.AAN:   .ASCII  \WRITE\                     ;
                                                                00081          .BLKB   3                           ;
                                        44  41  45  52          00084 P.AAO:   .ASCII  \READ\                      ;
                                45  54  49  52  57  4F  4E      00088 P.AAP:   .ASCII  \NOWRITE\                   ;
                                                                0008F          .BLKB   1                           ;
                                    44  41  45  52  4F  4E      00090 P.AAQ:   .ASCII  \NOREAD\                    ;
                                                                00096          .BLKB   2                           ;
45  54  45  4C  45  44  5F  4E  4F  5F  45  53  41  52  45      00098 P.AAS:   .ASCII  \ERASE_ON_DELETE\<0>         ;
                                                            00  000A7                                              ;
                                                    010E000F    000A8 P.AAR:   .LONG   17694735                     ;
                                                    00000000'   000AC          .ADDRESS P.AAS                      ;
            00  00  00  4E  4F  49  53  4E  45  54  58  45      000B0 P.AAU:   .ASCII  \EXTENSION\<0><0><0>          ;
```

```
                                                        010E0009  000BC P.AAT:   .LONG    17694729
                                                        00000000' 000C0        .ADDRESS P.AAU
                  00  00  00  4E  4F  49  53  4E  45  54  58  45  000C4 P.AAW:   .ASCII   \EXTENSION\<0><0><0>
                                                        010E0009  000D0 P.AAV:   .LONG    17694729
                                                        00000000' 000D4        .ADDRESS P.AAW
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  000D8 P.AAY:   .ASCII   \FILE_PROTECTION\<0>
                                                            00  000E7
                                                        010E000F  000E8 P.AAX:   .LONG    17694735
                                                        00000000' 000EC        .ADDRESS P.AAY
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  000F0 P.ABA:   .ASCII   \FILE_PROTECTION.SYSTEM\<0><0>
                        00  00  4D  45  54  53  59  53  2E  000FF
                                                        010E0016  00108 P.AAZ:   .LONG    17694742
                                                        00000000' 0010C        .ADDRESS P.ABA
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00110 P.ABC:   .ASCII   \FILE_PROTECTION.SYSTEM\<0><0>
                        00  00  4D  45  54  53  59  53  2E  0011F
                                                        010E0016  00128 P.ABB:   .LONG    17694742
                                                        00000000' 0012C        .ADDRESS P.ABC
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00130 P.ABE:   .ASCII   \FILE_PROTECTION.OWNER\<0><0><0>
                        00  00  00  52  45  4E  57  4F  2E  0013F
                                                        010E0015  00148 P.ABD:   .LONG    17694741
                                                        00000000' 0014C        .ADDRESS P.ABE
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00150 P.ABG:   .ASCII   \FILE_PROTECTION.OWNER\<0><0><0>
                        00  00  00  52  45  4E  57  4F  2E  0015F
                                                        010E0015  00168 P.ABF:   .LONG    17694741
                                                        00000000' 0016C        .ADDRESS P.ABG
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00170 P.ABI:   .ASCII   \FILE_PROTECTION.GROUP\<0><0><0>
                        00  00  00  50  55  4F  52  47  2E  0017F
                                                        010E0015  00188 P.ABH:   .LONG    17694741
                                                        00000000' 0018C        .ADDRESS P.ABI
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00190 P.ABK:   .ASCII   \FILE_PROTECTION.GROUP\<0><0><0>
                        00  00  00  50  55  4F  52  47  2E  0019F
                                                        010E0015  001A8 P.ABJ:   .LONG    17694741
                                                        00000000' 001AC        .ADDRESS P.ABK
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  001B0 P.ABM:   .ASCII   \FILE_PROTECTION.WORLD\<0><0><0>
                        00  00  00  44  4C  52  4F  57  2E  001BF
                                                        010E0015  001C8 P.ABL:   .LONG    17694741
                                                        00000000' 001CC        .ADDRESS P.ABM
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  001D0 P.ABO:   .ASCII   \FILE_PROTECTION.WORLD\<0><0><0>
                        00  00  00  44  4C  52  4F  57  2E  001DF
                                                        010E0015  001E8 P.ABN:   .LONG    17694741
                                                        00000000' 001EC        .ADDRESS P.ABO
49  4B  52  41  4D  5F  52  45  54  41  57  48  47  49  48  001F0 P.ABQ:   .ASCII   \HIGHWATER_MARKING\<0><0><0>
                                                    00  00  00  47  4E  001FF
                                                        010E0011  00204 P.ABP:   .LONG    17694737
                                                        00000000' 00208        .ADDRESS P.ABQ
                              00  00  00  4C  45  42  41  4C  0020C P.ABS:   .ASCII   \LABEL\<0><0><0>
                                                        010E0005  00214 P.ABR:   .LONG    17694725
                                                        00000000' 00218        .ADDRESS P.ABS
                              00  00  00  4C  45  42  41  4C  0021C P.ABU:   .ASCII   \LABEL\<0><0><0>
                                                        010E0005  00224 P.ABT:   .LONG    17694725
                                                        00000000' 00228        .ADDRESS P.ABU
                                                    00  47  4F  4C  0022C P.ABW:   .ASCII   \LOG\<0>
                                                        010E0003  00230 P.ABV:   .LONG    17694723
                                                        00000000' 00234        .ADDRESS P.ABW
54  41  43  49  46  49  52  45  56  5F  54  4E  55  4F  4D  00238 P.ABY:   .ASCII   \MOUNT_VERIFICATION\<0><0>
                                                    00  00  4E  4F  49  00247
                                                        010E0012  0024C P.ABX:   .LONG    17694738
```

```
                                                00000000'  00250                .ADDRESS P.ABY
                00  00  00  43  49  55  5F  52  45  4E  57  4F  00254 P.ACA:    .ASCII  \OWNER_UIC\<0><0><0>
                                                010E0009   00260 P.ABZ:    .LONG   17694729
                                                00000000'  00264                .ADDRESS P.ACA
                00  00  00  43  49  55  5F  52  45  4E  57  4F  00268 P.ACC:    .ASCII  \OWNER_UIC\<0><0><0>
                                                010E0009   00274 P.ACB:    .LONG   17694729
                                                00000000'  00278                .ADDRESS P.ACC
                                            0304  0004     0027C P.ACD:    .WORD   4, 772
                                                00000000G  00280                .ADDRESS UIC_VALUE
                                    00000000    00000000   00284                .LONG   0, 0
                00  00  4E  4F  49  54  43  45  54  4F  52  50  0028C P.ACF:    .ASCII  \PROTECTION\<0><0>
                                                010E000A   00298 P.ACE:    .LONG   17694730
                                                00000000'  0029C                .ADDRESS P.ACF
54  53  59  53  2E  4E  4F  49  54  43  45  54  4F  52  50  002A0 P.ACH:    .ASCII  \PROTECTION.SYSTEM\<0><0><0>
                                        00  00  00  4D  45  002AF
                                                010E0011   002B4 P.ACG:    .LONG   17694737
                                                00000000'  002B8                .ADDRESS P.ACH
54  53  59  53  2E  4E  4F  49  54  43  45  54  4F  52  50  002BC P.ACJ:    .ASCII  \PROTECTION.SYSTEM\<0><0><0>
                                        00  00  00  4D  45  002CB
                                                010E0011   002D0 P.ACI:    .LONG   17694737
                                                00000000'  002D4                .ADDRESS P.ACJ
45  4E  57  4F  2E  4E  4F  49  54  43  45  54  4F  52  50  002D8 P.ACL:    .ASCII  \PROTECTION.OWNER\
                                                        52  002E7
                                                010E0010   002E8 P.ACK:    .LONG   17694736
                                                00000000'  002EC                .ADDRESS P.ACL
45  4E  57  4F  2E  4E  4F  49  54  43  45  54  4F  52  50  002F0 P.ACN:    .ASCII  \PROTECTION.OWNER\
                                                        52  002FF
                                                010E0010   00300 P.ACM:    .LONG   17694736
                                                00000000'  00304                .ADDRESS P.ACN
55  4F  52  47  2E  4E  4F  49  54  43  45  54  4F  52  50  00308 P.ACP:    .ASCII  \PROTECTION.GROUP\
                                                        50  00317
                                                010E0010   00318 P.ACO:    .LONG   17694736
                                                00000000'  0031C                .ADDRESS P.ACP
55  4F  52  47  2E  4E  4F  49  54  43  45  54  4F  52  50  00320 P.ACR:    .ASCII  \PROTECTION.GROUP\
                                                        50  0032F
                                                010E0010   00330 P.ACQ:    .LONG   17694736
                                                00000000'  00334                .ADDRESS P.ACR
4C  52  4F  57  2E  4E  4F  49  54  43  45  54  4F  52  50  00338 P.ACT:    .ASCII  \PROTECTION.WORLD\
                                                        44  00347
                                                010E0010   00348 P.ACS:    .LONG   17694736
                                                00000000'  0034C                .ADDRESS P.ACT
4C  52  4F  57  2E  4E  4F  49  54  43  45  54  4F  52  50  00350 P.ACV:    .ASCII  \PROTECTION.WORLD\
                                                        44  0035F
                                                010E0010   00360 P.ACU:    .LONG   17694736
                                                00000000'  00364                .ADDRESS P.ACV
                                    00  44  4C  49  55  42  45  52  00368 P.ACX:    .ASCII  \REBUILD\<0>
                                                010E0007   00370 P.ACW:    .LONG   17694727
                                                00000000'  00374                .ADDRESS P.ACX
                00  00  00  4E  4F  49  54  4E  45  54  45  52  00378 P.ACZ:    .ASCII  \RETENTION\<0><0><0>
                                                010E0009   00384 P.ACY:    .LONG   17694729
                                                00000000'  00388                .ADDRESS P.ACZ
                00  00  00  4E  4F  49  54  4E  45  54  45  52  0038C P.ADB:    .ASCII  \RETENTION\<0><0><0>
                                                010E0009   00398 P.ADA:    .LONG   17694729
                                                00000000'  0039C                .ADDRESS P.ADB
                00  00  00  4E  4F  49  54  4E  45  54  45  52  003A0 P.ADD:    .ASCII  \RETENTION\<0><0><0>
                                                010E0009   003AC P.ADC:    .LONG   17694729
                                                00000000'  003B0                .ADDRESS P.ADD
```

```
                    00  00  44  41  4F  4C  4E  55  003B4 P.ADF:   .ASCII  \UNLOAD\<0><0>                          ;
                                                010E0006  003BC P.ADE:   .LONG   17694726                          ;
                                                00000000' 003C0          .ADDRESS P.ADF                         ;
        00  00  00  45  4D  41  4E  5F  52  45  53  55  003C4 P.ADH:   .ASCII  \USER_NAME\<0><0><0>                 ;
                                                010E0009  003D0 P.ADG:   .LONG   17694729                          ;
                                                00000000' 003D4          .ADDRESS P.ADH                         ;
        00  00  00  45  4D  41  4E  5F  52  45  53  55  003D8 P.ADJ:   .ASCII  \USER_NAME\<0><0><0>                 ;
                                                010E0009  003E4 P.ADI:   .LONG   17694729                          ;
                                                00000000' 003E8          .ADDRESS P.ADJ                         ;
                    00  53  57  4F  44  4E  49  57  003EC P.ADL:   .ASCII  \WINDOWS\<0>                           ;
                                                010E0007  003F4 P.ADK:   .LONG   17694727                          ;
                                                00000000' 003F8          .ADDRESS P.ADL                         ;
                    00  53  57  4F  44  4E  49  57  003FC P.ADN:   .ASCII  \WINDOWS\<0>                           ;
                                                010E0007  00404 P.ADM:   .LONG   17694727                          ;
                                                00000000' 00408          .ADDRESS P.ADN                         ;

                                                                 SETPRO_MASK=       FPROT_VALUE+2
                                                                 SETPRO_PROT=       FPROT_VALUE
                                                                 SETPRO_MASK=       VPROT_VALUE+2
                                                                 SETPRO_PROT=       VPROT_VALUE
                                                                 .EXTRN  SYS$SETPRV, SYS$GETJPIW

                                                                 .PSECT  $CODE$,NOWRT,2

                                                    0FFC 00000 GET_QUALS:
                                                                 .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11   ; 0431
                                    5B 00000000G   00   9E 00002        MOVAB   CLI$GET_VALUE, R11             ;
                                    5A 00000000G   00   9E 00009        MOVAB   CLI$PRESENT, R10               ;
                                    59 00000000'   EF   9E 00010        MOVAB   FLAGS, R9                      ;
                                    58 00000000'   EF   9E 00017        MOVAB   RETMIN_VALUE, R8               ;
                                    57 00000000'   EF   9E 0001E        MOVAB   P.AAD, R7                      ;
                                    5E             28   C2 00025        SUBL2   #40, SP                        ;
                                20  AE 020E0000    8F   D0 00028        MOVL    #34471936, DESC                ; 0448
                                       24  AE             D4 00030        CLRL    DESC+4                         ;
                                                   57   DD 00033        PUSHL   R7                             ; 0453
                                    6A             01   FB 00035        CALLS   #1, CLI$PRESENT                ;
                                    62             50   E9 00038        BLBC    R0, 7$                         ;
                                    69             02   88 0003B        BISB2   #2, FLAGS                      ; 0457
                                       18  AE          9F 0003E        PUSHAB  PRIVS                          ; 0466
                                                   01   DD 00041        PUSHL   #1                             ;
                                    7E             01   7D 00043        MOVQ    #1, -(SP)                      ;
                            00000000G   00         04   FB 00046        CALLS   #4, SYS$SETPRV                 ;
                                    56             50   D0 0004D        MOVL    R0, STATUS                     ;
                                    03             56   E8 00050        BLBS    STATUS, 1$                     ;
                                                04C9   31 00053        BRW     48$                            ;
                09      1A  AE                     02   E0 00056 1$:     BBS     #2, PRIVS+2, 2$                ; 0472
                                            00000000G  9F   DD 0005B        PUSHL   #SET$_OPERREQ                  ; 0475
                                                04BD   31 00061        BRW     49$                            ;
                                EC  A8                 03   D0 00064 2$:     MOVL    #3, ACC_VALUE          ; 0482
                                       20  AE          9F 00068        PUSHAB  DESC                           ; 0487
                                       10  A7          9F 0006B        PUSHAB  P.AAF                          ;
                                    68             02   FB 0006E        CALLS   #2, CLI$GET_VALUE              ;
                                    29             50   E9 00071        BLBC    R0, 7$                         ;
                                           EC  A8      9F 00074        PUSHAB  ACC_VALUE                      ; 0490
                                           28  AE      DD 00077        PUSHL   DESC+4                         ; 0491
                                    7E             28  AE 3C 0007A     MOVZWL  DESC, -(SP)                    ; 0490
                            00000000G   00         03   FB 0007E        CALLS   #3, LIB$CVT_DTB                ;
```

```
                          03              50  E8 00085        BLBS    R0, 4$                         ;
                                    04F4  31 00088 3$:       BRW     53$                            ;
                          50        EC    A8  D0 0008B 4$:    MOVL    ACC_VALUE, R0                  ; 0498
                                          03  18 0008F        BGEQ    6$                             ;
                                    050F  31 00091 5$:       BRW     57$                            ;
                000000FF  8F              50  D1 00094 6$:    CMPL    R0, #255                       ; 0499
                                          F4  14 0009B        BGTR    5$                             ;
                                    24    A7  9F 0009D 7$:    PUSHAB  P.AAH                          ; 0511
                          6A              01  FB 000A0        CALLS   #1, CLI$PRESENT                ;
                          5D              50  E9 000A3        BLBC    R0, 12$                        ;
                          69              04  88 000A6        BISB2   #4, FLAGS                      ; 0514
                                    20    AE  9F 000A9        PUSHAB  DESC                           ; 0515
                                    38    A7  9F 000AC        PUSHAB  P.AAJ                          ;
                          6B              02  FB 000AF        CALLS   #2, CLI$GET_VALUE              ;
                          06              50  E8 000B2        BLBS    R0, 8$                         ;
                      04  A9              04  88 000B5        BISB2   #4, DFLAGS                     ; 0517
                                          48  11 000B9        BRB     12$                            ;
                                    20    AE  9F 000BB 8$:    PUSHAB  DESC                           ; 0519
                                    4C    A7  9F 000BE        PUSHAB  P.AAL                          ;
                          6B              02  FB 000C1        CALLS   #2, CLI$GET_VALUE              ;
                          3C              50  E9 000C4        BLBC    R0, 12$                        ;
                          54        20    AE  3C 000C7        MOVZWL  DESC, R4                       ; 0521
        54   A7           24  BE          54  29 000CB        CMPC3   R4, @DESC+4, P.AAN             ;
                          06              12 000D1            BNEQ    9$                             ;
                      04  A9              04  88 000D3        BISB2   #4, DFLAGS                     ; 0523
                                          E2  11 000D7        BRB     8$                             ;
        5C   A7           24  BE          54  29 000D9 9$:    CMPC3   R4, @DESC+4, P.AAO             ; 0524
                                          06  12 000DF        BNEQ    10$                            ;
                      04  A9              02  88 000E1        BISB2   #2, DFLAGS                     ; 0526
                                          D4  11 000E5        BRB     8$                             ;
        60   A7           24  BE          54  29 000E7 10$:   CMPC3   R4, @DESC+4, P.AAP             ; 0527
                                          06  12 000ED        BNEQ    11$                            ;
                      04  A9              10  88 000EF        BISB2   #16, DFLAGS                    ; 0529
                                          C6  11 000F3        BRB     8$                             ;
        68   A7           24  BE          54  29 000F5 11$:   CMPC3   R4, @DESC+4, P.AAQ             ; 0530
                                          8B  12 000FB        BNEQ    3$                             ;
                      04  A9              08  88 000FD        BISB2   #8, DFLAGS                     ; 0532
                                          B8  11 00101        BRB     8$                             ;
                                  0080    C7  9F 00103 12$:   PUSHAB  P.AAR                          ; 0544
                          6A              01  FB 00107        CALLS   #1, CLI$PRESENT                ;
                          56              50  D0 0010A        MOVL    R0, STATUS                     ;
                00000000G 8F              56  D1 0010D        CMPL    STATUS, #CLI$_ABSENT           ; 0545
                                          0A  13 00114        BEQL    13$                            ;
                      01  A9              10  88 00116        BISB2   #16, FLAGS+1                   ; 0548
 01    A9             01  05              56  F0 0011A        INSV    STATUS, #5, #1, FLAGS+1        ; 0549
                                  0094    C7  9F 00120 13$:   PUSHAB  P.AAT                          ; 0555
                          6A              01  FB 00124        CALLS   #1, CLI$PRESENT                ;
                          46              50  E9 00127        BLBC    R0, 17$                        ;
                          69              08  88 0012A        BISB2   #8, FLAGS                      ; 0558
                00000000G 00              05  D0 0012D        MOVL    #5, EXTE_VALUE                 ; 0559
                                    20    AE  9F 00134        PUSHAB  DESC                           ; 0560
                                  00A8    C7  9F 00137        PUSHAB  P.AAV                          ;
                          6B              02  FB 0013B        CALLS   #2, CLI$GET_VALUE              ;
                          2F              50  E9 0013E        BLBC    R0, 17$                        ;
                            00000000G     00  9F 00141        PUSHAB  EXTE_VALUE                     ; 0563
                                    28    AE  DD 00147        PUSHL   DESC+4                         ; 0564
                          7E        28    AE  3C 0014A        MOVZWL  DESC, -(SP)                    ; 0563
```

```
          00000000G  00              03  FB 0014E         CALLS   #3, LIB$CVT_DTB                ;
                     03              50  E8 00155         BLBS    R0, 14$                        ;
                                   0424  31 00158         BRW     53$                            ;
                  50 00000000G       00  D0 0015B 14$:    MOVL    EXTE_VALUE, R0                 ; 0571
                                     03  18 00162         BGEQ    16$                            ;
                                   043C  31 00164 15$:    BRW     57$                            ;
          0000FFFF   8F              50  D1 00167 16$:    CMPL    R0, #65535                     ; 0572
                                     F4  14 0016E         BGTR    15$                            ;
                             00C0    C7  9F 00170 17$:    PUSHAB  P.AAX                          ; 0584
                     6A              01  FB 00174         CALLS   #1, CLI$PRESENT                ;
                     03              50  E8 00177         BLBS    R0, 18$                        ;
                                   00B7  31 0017A         BRW     22$                            ;
                     69              10  88 0017D 18$:    BISB2   #16, FLAGS                     ; 0591
                               F0    A8  D4 00180         CLRL    FPROT_VALUE                    ; 0592
                             00E0    C7  9F 00183         PUSHAB  P.AAZ                          ; 0594
                     6A              01  FB 00187         CALLS   #1, CLI$PRESENT                ;
                     1F              50  E9 0018A         BLBC    R0, 19$                        ;
                 F2  A8              0F  88 0018D         BISB2   #15, SETPRO_MASK               ; 0597
                               20    AE  9F 00191         PUSHAB  DESC                           ; 0598
                             0100    C7  9F 00194         PUSHAB  P.ABB                          ;
                     6B              02  FB 00198         CALLS   #2, CLI$GET_VALUE              ;
                     0E              50  E9 0019B         BLBC    R0, 19$                        ;
                               20    AE  9F 0019E         PUSHAB  DESC                           ; 0599
          00000000V  EF              01  FB 001A1         CALLS   #1, PARSE_CLASS                ;
                 F0  A8              50  B0 001A8         MOVW    R0, SETPRO_PROT                ;
                             0120    C7  9F 001AC 19$:    PUSHAB  P.ABD                          ; 0601
                     6A              01  FB 001B0         CALLS   #1, CLI$PRESENT                ;
                     23              50  E9 001B3         BLBC    R0, 20$                        ;
                 F2  A8        F0    8F  88 001B6         BISB2   #240, SETPRO_MASK              ; 0604
                               20    AE  9F 001BB         PUSHAB  DESC                           ; 0605
                             0140    C7  9F 001BE         PUSHAB  P.ABF                          ;
                     6B              02  FB 001C2         CALLS   #2, CLI$GET_VALUE              ;
                     11              50  E9 001C5         BLBC    R0, 20$                        ;
                               20    AE  9F 001C8         PUSHAB  DESC                           ; 0606
          00000000V  EF              01  FB 001CB         CALLS   #1, PARSE_CLASS                ;
                     50              10  C4 001D2         MULL2   #16, R0                        ;
                 F0  A8              50  A8 001D5         BISW2   R0, SETPRO_PROT                ;
                             0160    C7  9F 001D9 20$:    PUSHAB  P.ABH                          ; 0608
                     6A              01  FB 001DD         CALLS   #1, CLI$PRESENT                ;
                     23              50  E9 001E0         BLBC    R0, 21$                        ;
                 F3  A8              0F  88 001E3         BISB2   #15, SETPRO_MASK+1             ; 0611
                               20    AE  9F 001E7         PUSHAB  DESC                           ; 0612
                             0180    C7  9F 001EA         PUSHAB  P.ABJ                          ;
                     6B              02  FB 001EE         CALLS   #2, CLI$GET_VALUE              ;
                     12              50  E9 001F1         BLBC    R0, 21$                        ;
                               20    AE  9F 001F4         PUSHAB  DESC                           ; 0613
          00000000V  EF              01  FB 001F7         CALLS   #1, PARSE_CLASS                ;
      50             50              08  78 001FE         ASHL    #8, R0, R0                     ;
                 F0  A8              50  A8 00202         BISW2   R0, SETPRO_PROT                ;
                             01A0    C7  9F 00206 21$:    PUSHAB  P.ABL                          ; 0615
                     6A              01  FB 0020A         CALLS   #1, CLI$PRESENT                ;
                     24              50  E9 0020D         BLBC    R0, 22$                        ;
                 F3  A8        F0    8F  88 00210         BISB2   #240, SETPRO_MASK+1            ; 0618
                               20    AE  9F 00215         PUSHAB  DESC                           ; 0619
                             01C0    C7  9F 00218         PUSHAB  P.ABN                          ;
                     6B              02  FB 0021C         CALLS   #2, CLI$GET_VALUE              ;
                     12              50  E9 0021F         BLBC    R0, 22$                        ;
```

```
                                    20  AE  9F 00222          PUSHAB  DESC                                    ; 0620
                00000000V           EF      01  FB 00225          CALLS   #1, PARSE_CLASS
                        50          50      0C  78 0022C          ASHL    #12, R0, R0
                           F0       A8      50  A8 00230          BISW2   R0, SETPRO_PROT
                                    01DC    C7  9F 00234 22$:     PUSHAB  P.ABP                           ; 0627
                                    6A      01  FB 00238          CALLS   #1, CLI$PRESENT
                                    56      50  D0 0023B          MOVL    R0, STATUS
                00000000G           8F      56  D1 0023E          CMPL    STATUS, #CLI$_ABSENT            ; 0628
                                            0E  13 00245          BEQL    23$
                           01       A9      40  8F 88 00247          BISB2   #64, FLAGS+1                ; 0631
                                    50      56  D2 0024C          MCOML   STATUS, R0                      ; 0632
01      A9              01                  07  50 F0 0024F          INSV    R0, #7, #1, FLAGS+1
                                    01EC    C7  9F 00255 23$:     PUSHAB  P.ABR                           ; 0638
                                    6A      01  FB 00259          CALLS   #1, CLI$PRESENT
                                    2E      50  E9 0025C          BLBC    R0, 25$
                                        20  AE  9F 0025F          PUSHAB  DESC                            ; 0640
                                    01FC    C7  9F 00262          PUSHAB  P.ABT
                                    6B      02  FB 00266          CALLS   #2, CLI$GET_VALUE
                                    21      50  E9 00269          BLBC    R0, 25$
                                    69      20  88 0026C          BISB2   #32, FLAGS                      ; 0643
                                    0C  20  AE  B1 0026F          CMPW    DESC, #12                       ; 0644
                                            03  1B 00273          BLEQU   24$
                                        0307    31 00275          BRW     53$
                           F4       A8  20  AE  3C 00278 24$:     MOVZWL  DESC, LABEL_VALUE               ; 0650
                           F8       A8  24  AE  D0 0027D          MOVL    DESC+4, LABEL_VALUE+4           ; 0651
                           20       AE  020E0000    8F  D0 00282          MOVL    #34471936, DESC         ; 0652
                                        24  AE  D4 0028A          CLRL    DESC+4
                                    0208    C7  9F 0028D 25$:     PUSHAB  P.ABV                           ; 0658
                                    6A      01  FB 00291          CALLS   #1, CLI$PRESENT
        69              01                  06  50 F0 00294          INSV    R0, #6, #1, FLAGS
                                    0224    C7  9F 00299          PUSHAB  P.ABX                           ; 0663
                                    6A      01  FB 0029D          CALLS   #1, CLI$PRESENT
                                    56      50  D0 002A0          MOVL    R0, STATUS
                00000000G           8F      56  D1 002A3          CMPL    STATUS, #CLI$_ABSENT            ; 0664
                                            0A  13 002AA          BEQL    26$
                              02    A9      01  88 002AC          BISB2   #1, FLAGS+2                     ; 0667
02      A9              01                  01  56 F0 002B0          INSV    STATUS, #1, #1, FLAGS+2         ; 0668
                                    0238    C7  9F 002B6 26$:     PUSHAB  P.ABZ                           ; 0674
                                    6A      01  FB 002BA          CALLS   #1, CLI$PRESENT
                                    45      50  E9 002BD          BLBC    R0, 29$
                                    69      80  8F 88 002C0          BISB2   #128, FLAGS                 ; 0677
                                        20  AE  9F 002C4          PUSHAB  DESC                            ; 0678
                                    024C    C7  9F 002C7          PUSHAB  P.ACB
                                    6B      02  FB 002CB          CALLS   #2, CLI$GET_VALUE
                                    24      50  E8 002CE          BLBS    R0, 28$
                                            7E  7C 002D1          CLRQ    -(SP)                           ; 0687
                                        20  AE  9F 002D3          PUSHAB  IOSB
                                    0254    C7  9F 002D6          PUSHAB  P.ACD
                                            7E  7C 002DA          CLRQ    -(SP)
                                            7E  D4 002DC          CLRL    -(SP)
                00000000G           00      07  FB 002DE          CALLS   #7, SYS$GETJPIW
                                    56      50  D0 002E5          MOVL    R0, STATUS
                                    07      56  E9 002E8          BLBC    STATUS, 27$                     ; 0688
                                    56  18  AE  3C 002EB          MOVZWL  IOSB, STATUS                    ; 0689
                                    13      56  E8 002EF          BLBS    STATUS, 29$                     ; 0690
                                        022A    31 002F2 27$:     BRW     48$                             ; 0693
                    00000000G       00  9F 002F5 28$:     PUSHAB  UIC_VALUE                               ; 0697
```

```
                        24 AE 9F 002FB          PUSHAB  DESC                                    ;
00000000G   00              02 FB 002FE          CALLS   #2, PARSE_UIC                           ;
                      0270  C7 9F 00305 29$:     PUSHAB  P.ACE                                   ; 0703
                        6A  01 FB 00309          CALLS   #1, CLI$PRESENT                         ;
                        03  50 E8 0030C          BLBS    R0, 30$                                 ;
                          00B8 31 0030F          BRW     34$                                     ;
                        01  A9 04 88 00312 30$:  BISB2   #4, FLAGS+1                            ; 0710
                        FC  A8 D4 00316          CLRL    VPROT_VALUE                             ; 0711
                      028C  C7 9F 00319          PUSHAB  P.ACG                                   ; 0713
                        6A  01 FB 0031D          CALLS   #1, CLI$PRESENT                         ;
                        1F  50 E9 00320          BLBC    R0, 31$                                 ;
                        FE  A8 0F 88 00323       BISB2   #15, SETPRO_MASK                        ; 0716
                        20  AE 9F 00327          PUSHAB  DESC                                    ; 0717
                      02A8  C7 9F 0032A          PUSHAB  P.ACI                                   ;
                        6B  02 FB 0032E          CALLS   #2, CLI$GET_VALUE                       ;
                        0E  50 E9 00331          BLBC    R0, 31$                                 ;
                        20  AE 9F 00334          PUSHAB  DESC                                    ; 0718
00000000V   EF              01 FB 00337          CALLS   #1, PARSE_CLASS                         ;
                        FC  A8 50 B0 0033E       MOVW    R0, SETPRO_PROT                         ;
                      02C0  C7 9F 00342 31$:     PUSHAB  P.ACK                                   ; 0720
                        6A  01 FB 00346          CALLS   #1, CLI$PRESENT                         ;
                        23  50 E9 00349          BLBC    R0, 32$                                 ;
                        FE  A8 F0 8F 88 0034C    BISB2   #240, SETPRO_MASK                       ; 0723
                        20  AE 9F 00351          PUSHAB  DESC                                    ; 0724
                      02D8  C7 9F 00354          PUSHAB  P.ACM                                   ;
                        6B  02 FB 00358          CALLS   #2, CLI$GET_VALUE                       ;
                        11  50 E9 0035B          BLBC    R0, 32$                                 ;
                        20  AE 9F 0035E          PUSHAB  DESC                                    ; 0725
00000000V   EF              01 FB 00361          CALLS   #1, PARSE_CLASS                         ;
                        50  10 C4 00368          MULL2   #16, R0                                 ;
                        FC  A8 50 A8 0036B       BISW2   R0, SETPRO_PROT                         ;
                      02F0  C7 9F 0036F 32$:     PUSHAB  P.ACO                                   ; 0727
                        6A  01 FB 00373          CALLS   #1, CLI$PRESENT                         ;
                        23  50 E9 00376          BLBC    R0, 33$                                 ;
                        FF  A8 0F 88 00379       BISB2   #15, SETPRO_MASK+1                      ; 0730
                        20  AE 9F 0037D          PUSHAB  DESC                                    ; 0731
                      0308  C7 9F 00380          PUSHAB  P.ACQ                                   ;
                        6B  02 FB 00384          CALLS   #2, CLI$GET_VALUE                       ;
                        12  50 E9 00387          BLBC    R0, 33$                                 ;
                        20  AE 9F 0038A          PUSHAB  DESC                                    ; 0732
00000000V   EF              01 FB 0038D          CALLS   #1, PARSE_CLASS                         ;
                50      50  08 78 00394          ASHL    #8, R0, R0                              ;
                        FC  A8 50 A8 00398       BISW2   R0, SETPRO_PROT                         ;
                      0320  C7 9F 0039C 33$:     PUSHAB  P.ACS                                   ; 0734
                        6A  01 FB 003A0          CALLS   #1, CLI$PRESENT                         ;
                        24  50 E9 003A3          BLBC    R0, 34$                                 ;
                        FF  A8 F0 8F 88 003A6    BISB2   #240, SETPRO_MASK+1                     ; 0737
                        20  AE 9F 003AB          PUSHAB  DESC                                    ; 0738
                      0338  C7 9F 003AE          PUSHAB  P.ACU                                   ;
                        6B  02 FB 003B2          CALLS   #2, CLI$GET_VALUE                       ;
                        12  50 E9 003B5          BLBC    R0, 34$                                 ;
                        20  AE 9F 003B8          PUSHAB  DESC                                    ; 0739
00000000V   EF              01 FB 003BB          CALLS   #1, PARSE_CLASS                         ;
                50      50  0C 78 003C2          ASHL    #12, R0, R0                             ;
                        FC  A8 50 A8 003C6       BISW2   R0, SETPRO_PROT                         ;
                      0348  C7 9F 003CA 34$:     PUSHAB  P.ACW                                   ; 0746
                        6A  01 FB 003CE          CALLS   #1, CLI$PRESENT                         ;
```

```
                          56          50  D0 003D1        MOVL    R0, STATUS                                 ;
               00000000G  8F          56  D1 003D4        CMPL    STATUS, #CLI$_ABSENT                       ; 0747
                                      0A  13 003DB        BEQL    35$                                        ;
                      02  A9          10  88 003DD        BISB2   #16, FLAGS+2                               ; 0750
     02   A9          01  05          56  F0 003E1        INSV    STATUS, #5, #1, FLAGS+2                    ; 0751
                                035C  C7  9F 003E7  35$:  PUSHAB  P.ACY                                      ; 0757
                          6A          01  FB 003EB        CALLS   #1, CLI$PRESENT                            ;
                          69          50  E9 003EE        BLBC    R0, 40$                                    ;
                      01  A9          01  88 003F1        BISB2   #1, FLAGS+1                                ; 0762
          08          00  6E          00  2C 003F5        MOVC5   #0, (SP), #0, #8, RETMIN_VALUE             ; 0764
                                      68     003FA                                                           ;
          08          00  6E          00  2C 003FB        MOVC5   #0, (SP), #0, #8, RETMAX_VALUE             ; 0765
                                  08  A8     00400                                                           ;
                                  20  AE  9F 00402        PUSHAB  DESC                                       ; 0770
                                0370  C7  9F 00405        PUSHAB  P.ADA                                      ;
                          6B          02  FB 00409        CALLS   #2, CLI$GET_VALUE                          ;
                          03          50  E8 0040C        BLBS    R0, 36$                                    ;
                                    016D  31 0040F        BRW     53$                                        ;
                                  18  AE  9F 00412  36$:  PUSHAB  TEMP_DESC                                  ; 0780
                                  24  AE  9F 00415        PUSHAB  DESC                                       ;
               00000000G  00          02  FB 00418        CALLS   #2, LIB$CVT_DTIME                          ;
                          56          50  D0 0041F        MOVL    R0, STATUS                                 ;
                          04          56  E8 00422        BLBS    STATUS, 37$                                ;
                                      58  DD 00425        PUSHL   R8                                         ; 0783
                                      28  11 00427        BRB     38$                                        ;
               68     18  AE          08  28 00429  37$:  MOVC3   #8, TEMP_DESC, RETMIN_VALUE                ; 0786
                                  20  AE  9F 0042E        PUSHAB  DESC                                       ; 0792
                                0384  C7  9F 00431        PUSHAB  P.ADC                                      ;
                          6B          02  FB 00435        CALLS   #2, CLI$GET_VALUE                          ;
                          21          50  E9 00438        BLBC    R0, 41$                                    ;
                                  18  AE  9F 0043B        PUSHAB  TEMP_DESC                                  ; 0795
                                  24  AE  9F 0043E        PUSHAB  DESC                                       ;
               00000000G  00          02  FB 00441        CALLS   #2, LIB$CVT_DTIME                          ;
                          56          50  D0 00448        MOVL    R0, STATUS                                 ;
                          06          56  E8 0044B        BLBS    STATUS, 39$                                ;
                                  08  A8  9F 0044E        PUSHAB  RETMAX_VALUE                               ; 0798
                                    012E  31 00451  38$:  BRW     54$                                        ;
     08   A8          18  AE          08  28 00454  39$:  MOVC3   #8, TEMP_DESC, RETMAX_VALUE                ; 0801
                                      56  11 0045A  40$:  BRB     46$                                        ; 0792
                          6E D71BC000 8F  D0 0045C  41$:  MOVL    #-686047232, ONE_WEEK                      ; 0811
                      04  AE    FA7F  8F  32 00463        CVTWL   #-1409, ONE_WEEK+4                         ;
     10   AE          68          68  C1 00469        ADDL3   RETMIN_VALUE, RETMIN_VALUE, DOUBLE         ; 0818
                      14  AE      04  A8  D0 0046E        MOVL    RETMIN_VALUE+4, DOUBLE                     ;
                      14  AE      14  AE  D8 00473        ADWC    DOUBLE, DOUBLE                             ;
     08   AE          68          6E  C1 00478        ADDL3   ONE_WEEK, RETMIN_VALUE, WEEK_PLUS          ; 0819
                      0C  AE      04  A8  D0 0047D        MOVL    RETMIN_VALUE+4, WEEK_PLUS                  ;
                      0C  AE      04  AE  D8 00482        ADWC    ONE_WEEK, WEEK_PLUS                        ;
                          50          01  CE 00487        MNEGL   #1, R0                                     ; 0820
                      0C  AE      14  AE  D1 0048A        CMPL    DOUBLE, WEEK_PLUS                          ;
                                      0F  19 0048F        BLSS    44$                                        ;
                                      09  14 00491        BGTR    42$                                        ;
                  08  AE          10  AE  D1 00493        CMPL    DOUBLE, WEEK_PLUS                          ;
                                      04  13 00498        BEQL    43$                                        ;
                                      04  1F 0049A        BLSSU   44$                                        ;
                                      50  D6 0049C  42$:  INCL    R0                                         ;
                                      50  D6 0049E  43$:  INCL    R0                                         ;
                                      50  D5 004A0  44$:  TSTL    R0                                         ;
```

```
                                           08  15 004A2        BLEQ     45$                                    ;
                08  A8      10  AE         08  28 004A4        MOVC3    #8, DOUBLE, RETMAX_VALUE               ; 0821
                                           06  11 004AA        BRB      46$                                    ;
                08  A8      08  AE         08  28 004AC 45$:   MOVC3    #8, WEEK_PLUS, RETMAX_VALUE            ; 0822
                                    0394   C7  9F 004B2 46$:   PUSHAB   P.ADE                                  ; 0829
                            6A             01  FB 004B6        CALLS    #1, CLI$PRESENT                        ;
                            56             50  D0 004B9        MOVL     R0, STATUS                             ;
                     00000000G  8F         56  D1 004BC        CMPL     STATUS, #CLI$_ABSENT                   ; 0830
                                           0A  13 004C3        BEQL     47$                                    ;
                            02  A9         04  88 004C5        BISB2    #4, FLAGS+2                            ; 0833
   02  A9            01         03         56  F0 004C9        INSV     STATUS, #3, #1, FLAGS+2                ; 0834
                                    03A8   C7  9F 004CF 47$:   PUSHAB   P.ADG                                  ; 0840
                                6A         01  FB 004D3        CALLS    #1, CLI$PRESENT                        ;
                                73         50  E9 004D6        BLBC     R0, 52$                                ;
                            01  A9         02  88 004D9        BISB2    #2, FLAGS+1                            ; 0843
                                      20   AE  9F 004DD        PUSHAB   DESC                                   ; 0844
                                    03BC   C7  9F 004E0        PUSHAB   P.ADI                                  ;
                                6B         02  FB 004E4        CALLS    #2, CLI$GET_VALUE                      ;
                                47         50  E8 004E7        BLBS     R0, 51$                                ;
                                50    10   AE  9E 004EA        MOVAB    JPI_LIST, $$ITMBLKPTR                  ; 0854
                                80 0202000C 8F D0 004EE        MOVL     #33685516, ($$ITMBLKPTR)+              ;
                                80    08   A9  9E 004F5        MOVAB    USER_LABEL, ($$ITMBLKPTR)+             ;
                                80    10   A8  9E 004F9        MOVAB    USER_VALUE, ($$ITMBLKPTR)+             ;
                                           80  D4 004FD        CLRL     ($$ITMBLKPTR)+                         ;
                                           7E  7C 004FF        CLRQ     -(SP)                                  ; 0856
                                      10   AE  9F 00501        PUSHAB   IOSB                                   ;
                                      1C   AE  9F 00504        PUSHAB   JPI_LIST                               ;
                                           7E  7C 00507        CLRQ     -(SP)                                  ;
                                           7E  D4 00509        CLRL     -(SP)                                  ;
                     00000000G  00         07  FB 0050B        CALLS    #7, SYS$GETJPIW                        ;
                                56         50  D0 00512        MOVL     R0, STATUS                             ;
                                07         56  E9 00515        BLBC     STATUS, 48$                            ; 0857
                                56    08   AE  3C 00518        MOVZWL   IOSB, STATUS                           ; 0858
                                0B         56  E8 0051C        BLBS     STATUS, 50$                            ; 0859
                                           56  DD 0051F 48$:   PUSHL    STATUS                                 ; 0862
                     00000000G  00         01  FB 00521 49$:   CALLS    #1, LIB$SIGNAL                         ;
                                           67  11 00528        BRB      55$                                    ; 0863
                            14  A8    08   A9  9E 0052A 50$:   MOVAB    USER_LABEL, USER_VALUE+4               ; 0865
                                           1B  11 0052F        BRB      52$                                    ; 0844
                            0C        20   AE  B1 00531 51$:   CMPW     DESC, #12                              ; 0869
                                           48  1A 00535        BGTRU    53$                                    ;
                            10  A8    20   AE  3C 00537        MOVZWL   DESC, USER_VALUE                       ; 0875
                            14  A8    24   AE  D0 0053C        MOVL     DESC+4, USER_VALUE+4                   ; 0876
                            20  AE 020E0000 8F D0 00541        MOVL     #34471936, DESC                        ; 0877
                                      24   AE  D4 00549        CLRL     DESC+4                                 ;
                                    03CC   C7  9F 0054C 52$:   PUSHAB   P.ADK                                  ; 0884
                                6A         01  FB 00550        CALLS    #1, CLI$PRESENT                        ;
                                67         50  E9 00553        BLBC     R0, 58$                                ;
                            01  A9         08  88 00556        BISB2    #8, FLAGS+1                            ; 0887
                            18  A8         07  D0 0055A        MOVL     #7, WINDOW_VALUE                       ; 0888
                                      20   AE  9F 0055E        PUSHAB   DESC                                   ; 0889
                                    03DC   C7  9F 00561        PUSHAB   P.ADM                                  ;
                                6B         02  FB 00565        CALLS    #2, CLI$GET_VALUE                      ;
                                52         50  E9 00568        BLBC     R0, 58$                                ;
                                      18   A8  9F 0056B        PUSHAB   WINDOW_VALUE                           ; 0892
                                      28   AE  DD 0056E        PUSHL    DESC+4                                 ; 0893
                                7E    28   AE  3C 00571        MOVZWL   DESC, -(SP)                            ; 0892
```

```
                00000000G  00          03  FB  00575          CALLS   #3, LIB$CVT_DTB                          ;
                           14          50  E8  0057C          BLBS    R0, 56$                                 ;
                                    20 AE  9F  0057F  53$:     PUSHAB  DESC                                    ; 0897
                                       01  DD  00582  54$:     PUSHL   #1                                      ;
                                007710FA 8F  DD  00584          PUSHL   #7803130                                ;
                00000000G  00           03  FB  0058A          CALLS   #3, LIB$SIGNAL                          ;
                                        2E  11  00591  55$:     BRB     59$                                     ; 0898
                           07       18  A8  D1  00593  56$:     CMPL    WINDOW_VALUE, #7                        ; 0900
                                        0A  19  00597          BLSS    57$                                     ;
                00000050   8F       18  A8  D1  00599          CMPL    WINDOW_VALUE, #80                       ; 0901
                                        1A  15  005A1          BLEQ    58$                                     ;
                                007711EA 8F  DD  005A3  57$:     PUSHL   #7803370                                ; 0904
                                    24  AE  9F  005A9          PUSHAB  DESC                                    ;
                                        01  DD  005AC          PUSHL   #1                                      ;
                                007710FA 8F  DD  005AE          PUSHL   #7803130                                ;
                00000000G  00           04  FB  005B4          CALLS   #4, LIB$SIGNAL                          ;
                                        04  11  005BB          BRB     59$                                     ; 0905
                           50           01  D0  005BD  58$:     MOVL    #1, R0                                  ; 0910
                                        04  005C0              RET                                             ;
                                        50  D4  005C1  59$:     CLRL    R0                                      ; 0911
                                            04  005C3          RET                                             ;

; Routine Size:  1476 bytes,    Routine Base:  $CODE$ + 0107
```

```
;  919     0912  1 ROUTINE process_volume_set (root_desc, original_rvn, max_rvn) : NOVALUE =
;  920     0913  2 BEGIN
;  921     0914  2
;  922     0915  2 !++
;  923     0916  2 !
;  924     0917  2 ! Find each volume in the volume set and modify it.
;  925     0918  2 !
;  926     0919  2 ! Inputs:
;  927     0920  2 !       root_desc - descriptor of root volume
;  928     0921  2 !       original_rvn - volume number of original volume
;  929     0922  2 !       max_rvn = highest volume number in set
;  930     0923  2 !
;  931     0924  2 ! Outputs:
;  932     0925  2 !       None.
;  933     0926  2 !
;  934     0927  2 !--
;  935     0928  2
;  936     0929  2 MAP
;  937     0930  2     root_desc : REF VECTOR;
;  938     0931  2
;  939     0932  2
;  940     0933  2 LOCAL
;  941     0934  2     status,
;  942     0935  2     status2,
;  943     0936  2     saved_flags,                        ! Saved original flags
;  944     0937  2     reduced_flags,                      ! Reduced flags
;  945     0938  2     this_rvn : volatile,
;  946     0939  2     iosb : VECTOR[4,WORD],              ! $GETDVI status block
;  947     0940  2     desc1 : VECTOR[2],                  ! Device descriptors
;  948     0941  2     desc2 : VECTOR[2],
;  949     0942  2     buffer1 : VECTOR[128,BYTE],         ! Device buffers
;  950     0943  2     buffer2 : VECTOR[128,BYTE],
;  951     0944  2     dvi_list : $ITMLST_DECL(ITEMS=2);   ! $GETDVI item list
;  952     0945  2
;  953     0946  2 !
;  954     0947  2 ! Do a little sneaky stuff first.  Transfer the root volume's name to the
;  955     0948  2 ! local descriptor.  Save the current flag settings, and calculate the
;  956     0949  2 ! flags for other volumes in this volume set.
;  957     0950  2 !
;  958     0951  2 desc1[0] = .root_desc[0];               ! Set up so we
;  959     0952  2 desc1[1] = buffer1;                     ! can loop easily
;  960     0953  2 desc2[1] = buffer2;
;  961     0954  2 CH$MOVE(.root_desc[0],
;  962     0955  2         .root_desc[1],
;  963     0956  2         buffer1);
;  964     0957  2
;  965     0958  2 saved_flags = .flags;                   ! Save original
;  966     0959  2 reduced_flags = .flags AND              ! The reduced set has
;  967     0960  3                 (1^qual_erase OR        ! only the ERASE
;  968     0961  3                  1^qual_erase_val OR    ! and
;  969     0962  3                  1^qual_fhw OR          ! HIGHWATER
;  970     0963  3                  1^qual_fhw_val);       ! qualifiers
;  971     0964  2
;  972     0965  2 !
;  973     0966  2 ! For each volume in the set, check to see if this is the original, or only
;  974     0967  2 ! one of the sister volumes, and set FLAGS accordingly.  To do this, we need
;  975     0968  2 ! to call $GETDVI to see what the volume number is.  But I'm getting ahead
```

```
;  976          0969  2 ! of myself...
;  977          0970  2 !
;  978          0971  2 WHILE true DO
;  979          0972  2     BEGIN
;  980          0973  2
;  981          0974  2 !
;  982          0975  2 ! Open a file to the disk.
;  983          0976  2 !
;  984          0977  3     fab[fab$b_fns] = .desc1[0];
;  985          0978  3     fab[fab$l_fna] = .desc1[1];
;  986          0979  4     IF NOT (status = $OPEN(FAB = fab))
;  987          0980  4     THEN SIGNAL(set$_writeerr,                  ! Error accessing
;  988          0981  4                 1,
;  989          0982  4                 desc1,                          ! this disk
;  990          0983  4                 .status)                        ! for this reason
;  991          0984  4     ELSE
;  992          0985  4         channel = .fab[fab$l_stv];
;  993          0986  3
;  994          0987  3 ! Get the next volume in the volume set, even if we can't use this one.
;  995          0988  3 ! If we can't even get to the next volume, then hang it up.
;  996          0989  3 !
;  997        P 0990  3     $ITMLST_INIT(ITMLST = dvi_list,             ! Set up DVI list
;  998        P 0991  3                  (ITMCOD = dvi$_volnumber,      ! want current
;  999        P 0992  3                   BUFADR = this_rvn),           ! volume number,
; 1000        P 0993  3                  (ITMCOD = dvi$_nextdevnam,     ! next volume
; 1001        P 0994  3                   BUFADR = buffer2,             ! name.
; 1002        P 0995  3                   BUFSIZ = %ALLOCATION(buffer2),
; 1003          0996  3                   RETLEN = desc2));
; 1004        P 0997  3     status2 = $GETDVIW(ITMLST = dvi_list,       ! Get the info.
; 1005        P 0998  3                        DEVNAM = desc1,
; 1006          0999  3                        IOSB   = iosb);
; 1007          1000  3     IF .status2                                 ! If an error at
; 1008          1001  3     THEN status2 = .iosb[0];                    ! this point, we
; 1009          1002  3     IF NOT .status2                             ! can't proceed,
; 1010          1003  3     THEN                                        ! since we don't
; 1011          1004  4         BEGIN                                   ! know which RVN
; 1012          1005  4         SIGNAL(set$_writeerr,                   ! we're playing with
; 1013          1006  4                1,
; 1014          1007  4                desc1,
; 1015          1008  4                .status2);
; 1016          1009  4         RETURN;
; 1017          1010  3         END;
; 1018          1011  3
; 1019          1012  3 !
; 1020          1013  3 ! If the OPEN was successful, then process this volume.
; 1021          1014  3 !
; 1022          1015  3     IF .status
; 1023          1016  3     THEN
; 1024          1017  4         BEGIN
; 1025          1018  4         IF .this_rvn NEQ .original_rvn          ! If not original disk,
; 1026          1019  4         THEN flags = .reduced_flags;            ! use reduced flags.
; 1027          1020  4         IF .flags NEQ 0                         ! If anything to set,
; 1028          1021  4         THEN process_one_volume(desc1);         ! do it.
; 1029          1022  4         flags = .saved_flags;                   ! Restore flags
; 1030          1023  3         END;
; 1031          1024  3
; 1032          1025  3     $DASSGN(CHAN = .channel);                   ! Deassign the channel
```

```
;  1033     1026  3
;  1034     1027  3      ! Perform volume rebuild, if requested.
;  1035     1028  3      !
;  1036     1029  4      IF .status AND ( .this_rvn EQL .original_rvn )
;  1037     1030  3      THEN
;  1038     1031  3          IF .flags[qual_rebuild] AND .flags[qual_rebuild_val]
;  1039     1032  3          THEN
;  1040     1033  4              BEGIN
;  1041     1034  4
;  1042     1035  4              EXTERNAL ROUTINE
;  1043     1036  4                  stand_alone_rebuild;    ! Perform volume rebuild
;  1044     1037  4
;  1045     1038  4              LOCAL chan: WORD;
;  1046     1039  4
;  1047     1040  4              status = $ASSIGN( DEVNAM=desc1, CHAN=chan );
;  1048     1041  4              IF NOT .status
;  1049     1042  4              THEN SIGNAL (set$_openout, 1, desc1, .status, 0);
;  1050     1043  4
;  1051     1044  4              stand_alone_rebuild( .chan );       ! Do the rebuild.
;  1052     1045  4
;  1053     1046  4              status = $DASSGN( CHAN=.chan );
;  1054     1047  4              IF NOT .status
;  1055     1048  4              THEN SIGNAL (set$_closeout, 1, desc1, .status, 0);
;  1056     1049  4
;  1057     1050  3              END;
;  1058     1051  3
;  1059     1052  3      IF .this_rvn EQL .max_rvn                   ! If end of volume
;  1060     1053  3      THEN EXITLOOP;                              ! set, leave
;  1061     1054  3
;  1062     1055  3          CH$MOVE(.desc2[0],                      ! Now switch to the
;  1063     1056  3                  buffer2,                        ! next volume in this
;  1064     1057  3                  buffer1);                       ! volume set.
;  1065     1058  2          END;
;  1066     1059  2
;  1067     1060  2      !
;  1068     1061  2      ! For this volume set, if the /LABEL flag was set, then we must also
;  1069     1062  2      ! modify [0,0]VOLSET.SYS on the root volume.  Note that ODS1 volumes
;  1070     1063  2      ! cannot be volume sets so will fail this test.
;  1071     1064  2      !
;  1072     1065  2      IF .max_rvn GTR 1
;  1073     1066  2      AND .flags[qual_label]
;  1074     1067  2      THEN modify_volset(.root_desc);
;  1075     1068  2
;  1076     1069  2      RETURN;
;  1077     1070  1      END;


                                                    .EXTRN  SYS$OPEN, SYS$DASSGN
                                                    .EXTRN  STAND_ALONE_REBUILD
                                                    .EXTRN  SYS$ASSIGN

                                    OFFC 00000  PROCESS_VOLUME_SET:
                                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 0912
                5B 00000000'   EF   9E 00002        MOVAB   FLAGS, R11                                  ;
                5E       FEC4  CE   9E 00009        MOVAB   -316(SP), SP                                ;
                56         04  AC   D0 0000E        MOVL    ROOT_DESC, R6                               ; 0951
```

```
                    EC   AD             66  D0  00012        MOVL    (R6), DESC1                              ;
                    F0   AD       FF64  CD  9E  00016        MOVAB   BUFFER1, DESC1+4                         ; 0952
                    E8   AD         20  AE  9E  0001C        MOVAB   BUFFER2, DESC2+4                         ; 0953
    FF64   CD       04   B6             66  28  00021        MOVC3   (R6), @4(R6), BUFFER1                    ; 0954
                         5A             6B  D0  00028        MOVL    FLAGS, SAVED_FLAGS                       ; 0958
           59            6B  FFFF0FFF   8F  CB  0002B        BICL3   #-61441, FLAGS, REDUCED_FLAGS            ; 0960
                  03BC   CB         EC  AD  90  00033  1$:   MOVB    DESC1, FAB+52                            ; 0977
                  03B4   CB         F0  AD  D0  00039        MOVL    DESC1+4, FAB+44                          ; 0978
                                  0388  CB  9F  0003F        PUSHAB  FAB                                      ; 0979
             00000000G   00             01  FB  00043        CALLS   #1, SYS$OPEN                             ;
                         57             50  D0  0004A        MOVL    R0, STATUS                               ;
                         16             57  E8  0004D        BLBS    STATUS, 2$                               ;
                                        57  DD  00050        PUSHL   STATUS                                   ; 0983
                                    EC  AD  9F  00052        PUSHAB  DESC1                                    ; 0980
                                        01  DD  00055        PUSHL   #1                                       ;
                             00000000G  8F  DD  00057        PUSHL   #SET$_WRITEERR                           ;
             00000000G   00             04  FB  0005D        CALLS   #4, LIB$SIGNAL                           ;
                                        07  11  00064        BRB     3$                                       ;
                  0224   CB       0394  CB  D0  00066  2$:   MOVL    FAB+12, CHANNEL                          ; 0985
                         50         04  AE  9E  0006D  3$:   MOVAB   DVI_LIST, $$ITMBLKPTR                    ; 0996
                         80  002E0004   8F  D0  00071        MOVL    #3014660, ($$ITMBLKPTR)+                 ;
                         80         FC  AD  9E  00078        MOVAB   THIS_RVN, ($$ITMBLKPTR)+                 ;
                                        80  D4  0007C        CLRL    ($$ITMBLKPTR)+                           ;
                         80  00340080   8F  D0  0007E        MOVL    #3408000, ($$ITMBLKPTR)+                 ;
                         80         20  AE  9E  00085        MOVAB   BUFFER2, ($$ITMBLKPTR)+                  ;
                         80         E4  AD  9E  00089        MOVAB   DESC2, ($$ITMBLKPTR)+                    ;
                                        80  D4  0008D        CLRL    ($$ITMBLKPTR)+                           ;
                                        7E  7C  0008F        CLRQ    -(SP)                                    ; 0999
                                        7E  D4  00091        CLRL    -(SP)                                    ;
                                    F4  AD  9F  00093        PUSHAB  IOSB                                     ;
                                    14  AE  9F  00096        PUSHAB  DVI_LIST                                 ;
                                    EC  AD  9F  00099        PUSHAB  DESC1                                    ;
                                        7E  7C  0009C        CLRQ    -(SP)                                    ;
             00000000G   00             08  FB  0009E        CALLS   #8, SYS$GETDVIW                          ;
                         58             50  D0  000A5        MOVL    R0, STATUS2                              ;
                         07             58  E9  000A8        BLBC    STATUS2, 4$                              ; 1000
                         58         F4  AD  3C  000AB        MOVZWL  IOSB, STATUS2                            ; 1001
                         15             58  E8  000AF        BLBS    STATUS2, 5$                              ; 1002
                                        58  DD  000B2  4$:   PUSHL   STATUS2                                  ; 1008
                                    EC  AD  9F  000B4        PUSHAB  DESC1                                    ; 1005
                                        01  DD  000B7        PUSHL   #1                                       ;
                             00000000G  8F  DD  000B9        PUSHL   #SET$_WRITEERR                           ;
             00000000G   00             04  FB  000BF        CALLS   #4, LIB$SIGNAL                           ;
                                            04  000C6        RET                                              ; 1004
                         1B             57  E9  000C7  5$:   BLBC    STATUS, 8$                               ; 1015
                    08   AC         FC  AD  D1  000CA        CMPL    THIS_RVN, ORIGINAL_RVN                   ; 1018
                                        03  13  000CF        BEQL    6$                                       ;
                         6B             59  D0  000D1        MOVL    REDUCED_FLAGS, FLAGS                     ; 1019
                                        6B  D5  000D4  6$:   TSTL    FLAGS                                    ; 1020
                                        0A  13  000D6        BEQL    7$                                       ;
                                    EC  AD  9F  000D8        PUSHAB  DESC1                                    ; 1021
             00000000V   EF             01  FB  000DB        CALLS   #1, PROCESS_ONE_VOLUME                   ;
                         6B             5A  D0  000E2  7$:   MOVL    SAVED_FLAGS, FLAGS                       ; 1022
                                  0224  CB  DD  000E5  8$:   PUSHL   CHANNEL                                  ; 1025
             00000000G   00             01  FB  000E9        CALLS   #1, SYS$DASSGN                           ;
                         6C             57  E9  000F0        BLBC    STATUS, 10$                              ; 1029
                    08   AC         FC  AD  D1  000F3        CMPL    THIS_RVN, ORIGINAL_RVN                   ;
```

```
                                         65  12 000F8           BNEQ     10$                                     :
                60        02  AB         04  E1 000FA           BBC      #4, FLAGS+2, 10$                        : 1031
                5B        02  AB         05  E1 000FF           BBC      #5, FLAGS+2, 10$                        :
                                         7E  7C 00104           CLRQ     -(SP)                                   : 1040
                                   08    AE  9F 00106           PUSHAB   CHAN                                    :
                                   EC    AD  9F 00109           PUSHAB   DESC1                                   :
                   00000000G  00         04  FB 0010C           CALLS    #4, SYS$ASSIGN                          :
                              57         50  D0 00113           MOVL     R0, STATUS                              :
                              16         57  E8 00116           BLBS     STATUS, 9$                              : 1041
                                         7E  D4 00119           CLRL     -(SP)                                   : 1042
                                         57  DD 0011B           PUSHL    STATUS                                  :
                                   EC    AD  9F 0011D           PUSHAB   DESC1                                   :
                                         01  DD 00120           PUSHL    #1                                      :
                             007710A2    8F  DD 00122           PUSHL    #7803042                                :
                   00000000G  00         05  FB 00128           CALLS    #5, LIB$SIGNAL                          :
                              7E         6E  3C 0012F 9$:       MOVZWL   CHAN, -(SP)                             : 1044
                   00000000G  00         01  FB 00132           CALLS    #1, STAND_ALONE_REBUILD                 :
                              7E         6E  3C 00139           MOVZWL   CHAN, -(SP)                             : 1046
                   00000000G  00         01  FB 0013C           CALLS    #1, SYS$DASSGN                          :
                              57         50  D0 00143           MOVL     R0, STATUS                              :
                              16         57  E8 00146           BLBS     STATUS, 10$                             : 1047
                                         7E  D4 00149           CLRL     -(SP)                                   : 1048
                                         57  DD 0014B           PUSHL    STATUS                                  :
                                   EC    AD  9F 0014D           PUSHAB   DESC1                                   :
                                         01  DD 00150           PUSHL    #1                                      :
                             0077105A    8F  DD 00152           PUSHL    #7802970                                :
                   00000000G  00         05  FB 00158           CALLS    #5, LIB$SIGNAL                          :
                          0C  AC   FC    AD  D1 0015F 10$:      CMPL     THIS_RVN, MAX_RVN                       : 1052
                                         0B  13 00164           BEQL     11$                                     :
      FF64    CD          20  AE   E4    AD  28 00166           MOVC3    DESC2, BUFFER2, BUFFER1                 : 1055
                                       FEC2  31 0016E           BRW      1$                                      : 0971
                              01   0C    AC  D1 00171 11$:      CMPL     MAX_RVN, #1                             : 1065
                                         0D  15 00175           BLEQ     12$                                     :
                09            6B         05  E1 00177           BBC      #5, FLAGS, 12$                          : 1066
                                         56  DD 0017B           PUSHL    R6                                      : 1067
                   00000000V  EF         01  FB 0017D           CALLS    #1, MODIFY_VOLSET                       :
                                             04 00184 12$:      RET                                              : 1070

; Routine Size:  389 bytes,    Routine Base:  $CODE$ + 06CB
```

```
;  1079     1071  1 ROUTINE process_one_volume (desc) : NOVALUE =
;  1080     1072  2 BEGIN
;  1081     1073  2
;  1082     1074  2 !++
;  1083     1075  2 !
;  1084     1076  2 ! Find each volume in the volume set and call the routines which
;  1085     1077  2 ! actually modify the data.
;  1086     1078  2 !
;  1087     1079  2 ! Inputs:
;  1088     1080  2 !       desc - address of volume descriptor
;  1089     1081  2 !
;  1090     1082  2 ! Outputs:
;  1091     1083  2 !       None.  The volumes and I/O database are modified.
;  1092     1084  2 !
;  1093     1085  2 !--
;  1094     1086  2
;  1095     1087  2 LOCAL
;  1096     1088  2     status,
;  1097     1089  2     vbn,                                ! Place to store vbn
;  1098     1090  2     ucb,                                ! Place to store UCB address
;  1099     1091  2     cluster;                            ! Cluster size of volume
;  1100     1092  2
;  1101     1093  2 IF NOT read_homeblock(cluster)          ! If can't read home block
;  1102     1094  2 THEN SIGNAL(set$_nohome)                ! then tell the user
;  1103     1095  2 ELSE
;  1104     1096  2     BEGIN
;  1105     1097  2     status = 0;                         ! Show that no homeblocks modified yet
;  1106     1098  2     INCR vbn FROM 2 TO .cluster*3 DO    ! Go thru all homeblocks on the volume
;  1107     1099  4         BEGIN
;  1108     1100  4 !
;  1109     1101  4 ! Call the routine that reads, modifies, and writes the homeblock.  If
;  1110     1102  4 ! successful, set STATUS = 1
;  1111     1103  4 !
;  1112     1104  4         IF set_home(.vbn, .desc)
;  1113     1105  4         THEN status = 1;
;  1114     1106  4         IF .ods1
;  1115     1107  4         THEN EXITLOOP;                  ! Finished for ODS1
;  1116     1108  3         END;
;  1117     1109  3 !
;  1118     1110  3 ! If STATUS = 1, then at least some of the homeblocks were good and were
;  1119     1111  3 ! modified.
;  1120     1112  3 !
;  1121     1113  3     IF .status THEN
;  1122     1114  4         BEGIN
;  1123     1115  4         ! So, go ahead and change the I/O database.
;  1124     1116  4         !
;  1125     1117  4         ucb = KERNEL_CALL (GET_CHANNELUCB, .channel);
;  1126     1118  4         KERNEL_CALL(SET_UCBVCB, .ucb);
;  1127     1119  4
;  1128     1120  4         IF .flags[qual_log]             ! If /LOG, tell user
;  1129     1121  4         THEN SIGNAL (set$_modified, 1, .desc);
;  1130     1122  3         END;
;  1131     1123  2     END;
;  1132     1124  2
;  1133     1125  2 RETURN;
;  1134     1126  1 END;
```

```
                                                          .EXTRN  SYS$CMKRNL

                                  00FC 00000 PROCESS_ONE_VOLUME:
                                                          .WORD   Save R2,R3,R4,R5,R6,R7                 ; 1071
                57 00000000G  9F   9E 00002               MOVAB   @#SYS$CMKRNL, R7                       ;
                56 00000000G  00   9E 00009               MOVAB   LIB$SIGNAL, R6                         ;
                55 00000000'  EF   9E 00010               MOVAB   ODS1, R5                               ;
                5E            04   C2 00017               SUBL2   #4, SP                                 ;
                              5E   DD 0001A               PUSHL   SP                                     ; 1093
       00000000V EF           01   FB 0001C               CALLS   #1, READ_HOMEBLOCK                     ;
                0A            50   E8 00023               BLBS    R0, 1$                                 ;
                   00000000G  8F   DD 00026               PUSHL   #SET$_NOHOME                           ; 1094
                66            01   FB 0002C               CALLS   #1, LIB$SIGNAL                         ;
                                   04 0002F               RET                                            ;
                              54   D4 00030 1$:           CLRL    STATUS                                 ; 1097
  53            6E            03   C5 00032               MULL3   #3, CLUSTER, R3                        ; 1098
                52            01   D0 00036               MOVL    #1, VBN                                ; 1104
                              15   11 00039               BRB     4$                                     ;
                          04  AC   DD 0003B 2$:           PUSHL   DESC                                   ;
                              52   DD 0003E               PUSHL   VBN                                    ;
       00000000V EF           02   FB 00040               CALLS   #2, SET_HOME                           ;
                03            50   E9 00047               BLBC    R0, 3$                                 ;
                54            01   D0 0004A               MOVL    #1, STATUS                             ; 1105
                04            65   E8 0004D 3$:           BLBS    ODS1, 5$                               ; 1106
  E7            52            53   F3 00050 4$:           AOBLEQ  R3, VBN, 2$                            ; 1098
                33            54   E9 00054 5$:           BLBC    STATUS, 6$                             ; 1113
                          03  A5   DD 00057               PUSHL   CHANNEL                                ; 1117
                              01   DD 0005A               PUSHL   #1                                     ;
                              5E   DD 0005C               PUSHL   SP                                     ;
                   00000000G  00   9F 0005E               PUSHAB  GET_CHANNELUCB                         ;
                67            04   FB 00064               CALLS   #4, SYS$CMKRNL                         ;
                              50   DD 00067               PUSHL   UCB                                    ; 1118
                              01   DD 00069               PUSHL   #1                                     ;
                              5E   DD 0006B               PUSHL   SP                                     ;
                   00000000V  EF   9F 0006D               PUSHAB  SET_UCBVCB                             ;
                67            04   FB 00073               CALLS   #4, SYS$CMKRNL                         ;
  0E      FDDF  C5            06   E1 00076               BBC     #6, FLAGS, 6$                          ; 1120
                          04  AC   DD 0007C               PUSHL   DESC                                   ; 1121
                              01   DD 0007F               PUSHL   #1                                     ;
                   00000000G  8F   DD 00081               PUSHL   #SET$_MODIFIED                         ;
                66            03   FB 00087               CALLS   #3, LIB$SIGNAL                         ;
                                   04 0008A 6$:           RET                                            ; 1126

; Routine Size:  139 bytes,    Routine Base:  $CODE$ + 0850
```

```
;  1136          1127  1 ROUTINE read_homeblock(cluster) =
;  1137          1128  1 !++
;  1138          1129  1 !
;  1139          1130  1 ! This routine reads the first good home block of the volume.
;  1140          1131  1 ! It uses $QIOW's because $READ finds the End-of-File block to be
;  1141          1132  1 ! zero in ODS1 initialized disks and thus will not try to read the home block.
;  1142          1133  1 ! In addition the cluster size and structure level are determined and stored.
;  1143          1134  1 !
;  1144          1135  1 ! Outputs:
;  1145          1136  1 !       cluster - cluster size
;  1146          1137  1 !       ods1 - 0 => ODS2
;  1147          1138  1 !              1 => ODS1
;  1148          1139  1 !
;  1149          1140  1 !--
;  1150          1141  2 BEGIN
;  1151          1142  2
;  1152          1143  2 LOCAL
;  1153          1144  2     desc : $BBLOCK[dsc$c_s_bln],          ! Descriptor for the FIB in $QIOW
;  1154          1145  2     fib  : $BBLOCK[fib$c_extdata],        ! File Information Block for $QIOW
;  1155          1146  2     atr  : BLOCKVECTOR[2,8,BYTE],         ! Attribute list for $QIOW
;  1156          1147  2     stablk : $BBLOCK[32],                 ! Where statistics block is stored after $QIOW
;  1157          1148  2     file_size : VECTOR[2,WORD],           ! The file size from statistics block
;  1158          1149  2     iosb : VECTOR[4,WORD],                ! Status block for the $QIOW
;  1159          1150  2     block,                                ! Temporary block count
;  1160          1151  2     status;                               ! Status
;  1161          1152  2
;  1162          1153  2 ! Before we can look at the homeblock we have to find how many blocks there
;  1163          1154  2 ! are (or the block number or the last block).  This is done by issuing a
;  1164          1155  2 ! $QIOW to get the statistics block.
;  1165          1156  2
;  1166          1157  2 desc[dsc$w_length] = fib$c_extdata;      ! Initialize descriptor pointing to
;  1167          1158  2 desc[dsc$a_pointer] = fib;               ! to the file info block
;  1168          1159  2
;  1169          1160  2 CH$FILL(0, fib$c_extdata, fib);          ! Zero the fib for new info
;  1170          1161  2
;  1171          1162  2 fib[fib$l_acctl] = fib$m_noread OR       ! Deny read and write access to others
;  1172          1163  2                   fib$m_nowrite;
;  1173          1164  2 fib[fib$w_fid_num] = .nam[nam$w_fid_num];
;  1174          1165  2 fib[fib$w_fid_seq] = .nam[nam$w_fid_seq];  ! Specify file identification
;  1175          1166  2 fib[fib$w_fid_rvn] = .nam[nam$w_fid_rvn];
;  1176          1167  2
;  1177          1168  2 atr[0,atr$w_type] = atr$c_statblk;       ! The attribute we want is the
;  1178          1169  2 atr[0,atr$w_size] = atr$s_statblk;       ! statistics block
;  1179          1170  2 atr[0,atr$l_addr] = stablk;             ! It goes into stablk
;  1180          1171  2 atr[1,0,0,32,0]   = 0;                  ! Indicate end of information
;  1181          1172  2
;  1182        P 1173  2 status = $QIOW (CHAN = .channel,        ! Access the statistics block
;  1183        P 1174  2                 FUNC = IO$_ACCESS,
;  1184        P 1175  2                 IOSB = iosb,
;  1185        P 1176  2                 P1   = desc,
;  1186          1177  2                 P5   = atr);
;  1187          1178  2 IF  .status THEN status = .iosb[0];      ! Check if everything Okay
;  1188          1179  2 IF NOT .status
;  1189          1180  2 THEN SIGNAL(.status)                    ! If not, tell user, go to end
;  1190          1181  2 ELSE                                    ! If okay
;  1191          1182  3 BEGIN
;  1192          1183  3     file_size[1] = .stablk[sbk$w_filesizh];  ! The file size is stored
```

```
;  1193          1184  3     file_size[0] = .stablk[sbk$w_filesizl];  ! backwards so invert
;  1194          1185  3
;  1195          1186  3
;  1196          1187  3 !
;  1197          1188  3 ! It is possible the homeblock exists so . . .
;  1198          1189  3 ! Keep reading until we get a block that reads without errors and meets the
;  1199          1190  3 ! criteria for a homeblock.
;  1200          1191  3 !
;  1201          1192  3     INCR block FROM 2 TO 100 DO
;  1202          1193  4         BEGIN
;  1203          1194  4         IF .block LEQ .file_size                ! If we have not passed the end of file
;  1204          1195  4         THEN
;  1205          1196  5         BEGIN
;  1206        P 1197  5             status = $QIOW (CHAN = .channel,        ! Read the virtual 'block'
;  1207        P 1198  5                             FUNC = IO$_READVBLK,
;  1208        P 1199  5                             IOSB = iosb,
;  1209        P 1200  5                             P1   = buffer,          ! Put it in 'buffer'
;  1210          1201  5                             P2   = 512,             ! Get a whole block
;  1211          1202  5                             P3   = .block);
;  1212          1203  5             IF  .status THEN status = .iosb[0];
;  1213          1204  5             IF NOT .status
;  1214          1205  6             THEN
;  1215          1206  6             BEGIN
;  1216          1207  6                 SIGNAL(.status);
;  1217          1208  6                 RETURN false;
;  1218          1209  5             END;
;  1219          1210  5             IF
;  1220          1211  5                 .buffer[hm2$b_strucLev] EQL 2 AND
;  1221          1212  5                 .buffer[hm2$l_altidxlbn] NEQ 0 AND
;  1222          1213  5                 .buffer[hm2$w_cluster] NEQ 0 AND
;  1223          1214  5                 .buffer[hm2$w_homevbn] NEQ 0 AND
;  1224          1215  5                 .buffer[hm2$w_alhomevbn] NEQ 0 AND
;  1225          1216  5                 .buffer[hm2$w_altidxvbn] NEQ 0 AND
;  1226          1217  5                 .buffer[hm2$w_ibmapvbn] NEQ 0 AND
;  1227          1218  5                 .buffer[hm2$l_ibmaplbn] NEQ 0 AND
;  1228          1219  5                 .buffer[hm2$l_maxfiles] NEQ 0 AND
;  1229          1220  5                 .buffer[hm2$w_ibmapsize] NEQ 0 AND
;  1230          1221  5                 .buffer[hm2$w_resfiles] NEQ 0 AND
;  1231          1222  5                 checksum2(buffer, $BYTEOFFSET(hm2$w_checksum1)) AND
;  1232          1223  5                 checksum2(buffer, $BYTEOFFSET(hm2$w_checksum2))
;  1233          1224  5             THEN
;  1234          1225  6             BEGIN
;  1235          1226  6                 ods1 = 0;                           ! This is an ODS2 volume
;  1236          1227  6                 .cluster = .buffer[hm2$w_cluster];  ! with this cluster size
;  1237          1228  6                 IF .flags[qual_access]  ! If /ACCESSED was specified,
;  1238          1229  7                 THEN                    ! compute the value to add
;  1239          1230  7                 BEGIN                   ! to the LRU value in the VCB
;  1240          1231  7                     acc_inc = 0;
;  1241          1232  7                     IF .acc_value GTR .buffer[hm2$b_lru_lim]
;  1242          1233  6                     THEN acc_inc = .acc_value - .buffer[hm2$b_lru_lim];
;  1243          1234  6                 END;
;  1244          1235  6                 RETURN true;
;  1245          1236  5             END
;  1246          1237  5                 ELSE IF
;  1247          1238  5                     .buffer[hm1$w_strucLev] EQL hm1$c_level1 AND
;  1248          1239  5                     .buffer[hm1$w_cluster] NEQ 0 AND
;  1249          1240  5                     .buffer[hm1$l_ibmaplbn] NEQ 0 AND
;  1250          1241  5                     .buffer[hm1$w_maxfiles] NEQ 0 AND
```

```
;  1250     1241  5                    .buffer[hm1$w_ibmapsize] NEQ 0 AND
;  1251     1242  5                    checksum2(buffer, $BYTEOFFSET(hm1$w_checksum1)) AND
;  1252     1243  5                    checksum2(buffer, $BYTEOFFSET(hm1$w_checksum2))
;  1253     1244  5                THEN
;  1254     1245  6                    BEGIN
;  1255     1246  6                    ods1 = 1;                               ! Volume is ODS1
;  1256     1247  6                    .cluster = 1;                           ! Dummy in a cluster size of 1
;  1257     1248  6                    IF .flags[qual_access]      ! If /ACCESSED was specified,
;  1258     1249  6                    THEN                                    ! compute the value to add
;  1259     1250  7                        BEGIN                               ! to the LRU value in the VCB
;  1260     1251  7                        acc_inc = 0;
;  1261     1252  7                        IF .acc_value GTR .buffer[hm1$b_lru_lim]
;  1262     1253  7                        THEN acc_inc = .acc_value - .buffer[hm1$b_lru_lim];
;  1263     1254  6                        END;
;  1264     1255  6                    RETURN true;
;  1265     1256  5                    END;
;  1266     1257  4                END;                                        ! End of read success block
;  1267     1258  3            END;                                            ! End of INC block
;  1268     1259  2        END;                                                ! End of file access block
;  1269     1260  2
;  1270     1261  2    ! If here, then no good homeblock was found. Return a value of FALSE to
;  1271     1262  2    ! show that.
;  1272     1263  2    !
;  1273     1264  2    RETURN false;
;  1274     1265  1    END;


                                    .EXTRN  SYS$QIOW

                            03FC 00000 READ_HOMEBLOCK:
                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9        ; 1127
                    59 00000000G    00  9E 00002    MOVAB   SYS$QIOW, R9
                    58 00000000'    EF  9E 00009    MOVAB   ACC_VALUE, R8
                    57 00000000G    00  9E 00010    MOVAB   CHECKSUM2, R7
                    56 00000000'    EF  9E 00017    MOVAB   BUFFER, R6
                    5E        9C    AE  9E 0001E    MOVAB   -100(SP), SP
                 5C AE            20  B0 00022    MOVW    #32, DESC                   ; 1157
                 60 AE         3C    AE  9E 00026    MOVAB   FIB, DESC+4             ; 1158
   20            00            6E    00  2C 0002B    MOVC5   #0, (SP), #0, #32, FIB  ; 1160
                                 3C    AE      00030
                 3C AE      0401    8F  3C 00032    MOVZWL  #1025, FIB              ; 1162
                 40 AE      032C    C6  D0 00038    MOVL    NAM+36, FIB+4           ; 1164
                 44 AE      0330    C6  B0 0003E    MOVW    NAM+40, FIB+8           ; 1166
                 2C AE  00090020    8F  D0 00044    MOVL    #589856, ATR            ; 1169
                 30 AE        0C    AE  9E 0004C    MOVAB   STABLK, ATR+4           ; 1170
                                34  AE  D4 00051    CLRL    ATR+8                   ; 1171
                                7E  D4 00054    CLRL    -(SP)                   ; 1177
                                30  AE  9F 00056    PUSHAB  ATR
                                7E  7C 00059    CLRQ    -(SP)
                                7E  D4 0005B    CLRL    -(SP)
                                70  AE  9F 0005D    PUSHAB  DESC
                                7E  7C 00060    CLRQ    -(SP)
                                24  AE  9F 00062    PUSHAB  IOSB
                                    32  DD 00065    PUSHL   #50
                              0204  C6  DD 00067    PUSHL   CHANNEL
                                7E  D4 0006B    CLRL    -(SP)
```

```
                69              0C  FB 0006D           CALLS   #12, SYS$QIOW                                   ;
                53              50  D0 00070           MOVL    R0, STATUS                                      ;
                07              53  E9 00073           BLBC    STATUS, 1$                                      ; 1178
                53      04      AE  3C 00076           MOVZWL  IOSB, STATUS                                    ;
                0C              53  E8 0007A           BLBS    STATUS, 2$                                      ; 1179
                                53  DD 0007D 1$:       PUSHL   STATUS                                          ; 1180
      00000000G 00              01  FB 0007F           CALLS   #1, LIB$SIGNAL                                  ;
                            0111  31 00086           BRW     8$                                              ;
            02  AE      10      AE  B0 00089 2$:       MOVW    STABLK+4, FILE_SIZE+2                           ; 1183
                6E      12      AE  B0 0008E           MOVW    STABLK+6, FILE_SIZE                             ; 1184
                52              02  D0 00092           MOVL    #2, BLOCK                                       ; 1193
                6E              52  D1 00095 3$:       CMPL    BLOCK, FILE_SIZE                                ;
                                03  15 00098           BLEQ    4$                                              ;
                            00F3  31 0009A           BRW     7$                                              ;
                                7E  7C 0009D 4$:       CLRQ    -(SP)                                           ; 1201
                                7E  D4 0009F           CLRL    -(SP)                                           ;
                                52  DD 000A1           PUSHL   BLOCK                                           ;
                7E      0200    8F  3C 000A3           MOVZWL  #512, -(SP)                                     ;
                                56  DD 000A8           PUSHL   R6                                              ;
                                7E  7C 000AA           CLRQ    -(SP)                                           ;
                            24  AE  9F 000AC           PUSHAB  IOSB                                            ;
                                31  DD 000AF           PUSHL   #49                                             ;
                            0204  C6  DD 000B1         PUSHL   CHANNEL                                         ;
                                7E  D4 000B5           CLRL    -(SP)                                           ;
                69              0C  FB 000B7           CALLS   #12, SYS$QIOW                                   ;
                53              50  D0 000BA           MOVL    R0, STATUS                                      ;
                BD              53  E9 000BD           BLBC    STATUS, 1$                                      ; 1202
                53      04      AE  3C 000C0           MOVZWL  IOSB, STATUS                                    ;
                B6              53  E9 000C4           BLBC    STATUS, 1$                                      ; 1203
                02      0D      A6  91 000C7           CMPB    BUFFER+13, #2                                   ; 1210
                                6C  12 000CB           BNEQ    5$                                              ;
                            08  A6  D5 000CD           TSTL    BUFFER+8                                        ; 1211
                                67  13 000D0           BEQL    5$                                              ;
                            0E  A6  B5 000D2           TSTW    BUFFER+14                                       ; 1212
                                62  13 000D5           BEQL    5$                                              ;
                            10  A6  B5 000D7           TSTW    BUFFER+16                                       ; 1213
                                5D  13 000DA           BEQL    5$                                              ;
                            12  A6  B5 000DC           TSTW    BUFFER+18                                       ; 1214
                                58  13 000DF           BEQL    5$                                              ;
                            14  A6  B5 000E1           TSTW    BUFFER+20                                       ; 1215
                                53  13 000E4           BEQL    5$                                              ;
                            16  A6  B5 000E6           TSTW    BUFFER+22                                       ; 1216
                                4E  13 000E9           BEQL    5$                                              ;
                            18  A6  D5 000EB           TSTL    BUFFER+24                                       ; 1217
                                49  13 000EE           BEQL    5$                                              ;
                            1C  A6  D5 000F0           TSTL    BUFFER+28                                       ; 1218
                                44  13 000F3           BEQL    5$                                              ;
                            20  A6  B5 000F5           TSTW    BUFFER+32                                       ; 1219
                                3F  13 000F8           BEQL    5$                                              ;
                            22  A6  B5 000FA           TSTW    BUFFER+34                                       ; 1220
                                3A  13 000FD           BEQL    5$                                              ;
                                3A  DD 000FF           PUSHL   #58                                             ; 1221
                                56  DD 00101           PUSHL   R6                                              ;
                67              02  FB 00103           CALLS   #2, CHECKSUM2                                   ;
                30              50  E9 00106           BLBC    R0, 5$                                          ;
                7E      01FE    8F  3C 00109           MOVZWL  #510, -(SP)                                     ; 1222
                                56  DD 0010E           PUSHL   R6                                              ;
```

```
                                67              02  FB 00110          CALLS   #2, CHECKSUM2                                    ;
                                23              50  E9 00113          BLBC    R0, 5$                                          ;
                          0201  C6              94 00116          CLRB    OD$1                                            ; 1225
                            04  BC          0E  A6  3C 0011A          MOVZWL  BUFFER+14, @CLUSTER                             ; 1226
                    68      E0  A6          01  E1  0011F          BBC     #1, FLAGS, 6$                                   ; 1227
                          0200  C6              94 00124          CLRB    ACC_INC                                         ; 1230
          68    45  A6          08              00  ED 00128          CMPZV   #0, #8, BUFFER+69, ACC_VALUE                    ; 1231
                                                5C  18 0012E          BGEQ    6$                                              ;
                0200  C6          68      45  A6  83 00130          SUBB3   BUFFER+69, ACC_VALUE, ACC_INC                   ; 1232
                                                53  11 00137          BRB     6$                                              ; 1234
                    0101  8F      0C      A6  B1 00139  5$:         CMPW    BUFFER+12, #257                                 ; 1237
                                                4F  12 0013F          BNEQ    7$                                              ;
                                          08    A6  B5 00141          TSTW    BUFFER+8                                        ; 1238
                                                4A  13 00144          BEQL    7$                                              ;
                                          02    A6  D5 00146          TSTL    BUFFER+2                                        ; 1239
                                                45  13 00149          BEQL    7$                                              ;
                                          06    A6  B5 0014B          TSTW    BUFFER+6                                        ; 1240
                                                40  13 0014E          BEQL    7$                                              ;
                                                66  B5 00150          TSTW    BUFFER                                          ; 1241
                                                3C  13 00152          BEQL    7$                                              ;
                                                3A  DD 00154          PUSHL   #58                                             ; 1242
                                                56  DD 00156          PUSHL   R6                                              ;
                                67              02  FB 00158          CALLS   #2, CHECKSUM2                                   ;
                                32              50  E9 0015B          BLBC    R0, 7$                                          ;
                                7E      01FE    8F  3C 0015E          MOVZWL  #510, -(SP)                                     ; 1243
                                                56  DD 00163          PUSHL   R6                                              ;
                                67              02  FB 00165          CALLS   #2, CHECKSUM2                                   ;
                                25              50  E9 00168          BLBC    R0, 7$                                          ;
                    0201  C6                   01  90 0016B          MOVB    #1, OD$1                                        ; 1246
                      04  BC                   01  D0 00170          MOVL    #1, @CLUSTER                                    ; 1247
                    13    E0  A6              01  E1 00174          BBC     #1, FLAGS, 6$                                   ; 1248
                                          0200  C6  94 00179          CLRB    ACC_INC                                         ; 1251
          68    2E  A6          08              00  ED 0017D          CMPZV   #0, #8, BUFFER+46, ACC_VALUE                    ; 1252
                                                07  18 00183          BGEQ    6$                                              ;
                0200  C6          68      2E  A6  83 00185          SUBB3   BUFFER+46, ACC_VALUE, ACC_INC                   ; 1253
                                          50    01  D0 0018C  6$:       MOVL    #1, R0                                          ; 1255
                                                04 0018F              RET                                                     ;
          FEFB      52          01 00000064    8F  F1 00190  7$:       ACBL    #100, #1, BLOCK, 3$                             ; 1191
                                                50  D4 0019A  8$:       CLRL    R0                                              ; 1265
                                                04 0019C              RET                                                     ;

; Routine Size:  413 bytes,    Routine Base:  $CODE$ + 08DB
```

```
;  1276          1266  1 ROUTINE parse_class (desc) =
;  1277          1267  2 BEGIN
;  1278          1268  2
;  1279          1269  2 !---
;  1280          1270  2 !
;  1281          1271  2 ! This routine parses one class of user (e.g. SYSTEM, OWNER, GROUP, WORLD)
;  1282          1272  2 ! to see what protection is allowed.  The value returned in the low 4 bits
;  1283          1273  2 ! is the protection code, with the bits set to reflect that access is
;  1284          1274  2 ! requested.  Note that this is exactly the opposite of what the system wants.
;  1285          1275  2 !
;  1286          1276  2 ! Inputs:
;  1287          1277  2 !
;  1288          1278  2 !       DESC - a descriptor pointing to the ASCII representation of the
;  1289          1279  2 !              protection desired
;  1290          1280  2 !
;  1291          1281  2 !---
;  1292          1282  2
;  1293          1283  2 MAP desc : REF $BBLOCK;
;  1294          1284  2
;  1295          1285  2 LOCAL
;  1296          1286  2     result,
;  1297          1287  2     string : REF VECTOR[,BYTE];          ! String pointer
;  1298          1288  2
;  1299          1289  2 !
;  1300          1290  2 ! Initially set the value to all zeros, no access
;  1301          1291  2 !
;  1302          1292  2 result = 0;
;  1303          1293  2
;  1304          1294  2 !
;  1305          1295  2 ! Scan for the occurrence of each keyletter, and, if it is there, set the
;  1306          1296  2 ! appropriate bit.
;  1307          1297  2 !
;  1308          1298  2 string = .desc[dsc$a_pointer];
;  1309          1299  2 INCR index FROM 0 to (.desc[dsc$w_length] -1) DO
;  1310          1300  2     BEGIN
;  1311          1301  2     IF .string[.index] EQL'R'
;  1312          1302  2     THEN result = .result OR %X'1'
;  1313          1303  2     ELSE IF .string[.index] EQL  'W'
;  1314          1304  2     THEN result = .result OR %X'2'
;  1315          1305  2     ELSE IF .string[.index] EQL 'E'
;  1316          1306  2           OR .string[.index] EQL 'P'
;  1317          1307  2     THEN result = .result OR %X'4'
;  1318          1308  2     ELSE IF .string[.index] EQL 'D'
;  1319          1309  2           OR .string[.index] EQL 'L'
;  1320          1310  2     THEN result = .result OR %X'8'
;  1321          1311  2     ELSE SIGNAL_STOP(cli$_ivprot);
;  1322          1312  2     END;
;  1323          1313  2
;  1324          1314  2 RETURN .result;
;  1325          1315  1 END;


                                003C 00000 PARSE_CLASS:
                                           .WORD   Save R2,R3,R4,R5                ; 1266
```

```
                                54  D4 00002          CLRL    RESULT                                  ; 1292
                            50  04  AC  D0 00004      MOVL    DESC, R0                                ; 1298
                            52  04  A0  D0 00008      MOVL    4(R0), STRING
                            55      60  3C 0000C      MOVZWL  (R0), R5                                ; 1299
                            53      01  CE 0000F      MNEGL   #1, INDEX                               ; 1301
                                    49  11 00012      BRB     8$
                            50    6342  9A 00014 1$:  MOVZBL  (INDEX)[STRING], R0
                        52  8F        50  91 00018      CMPB    R0, #82
                                      05  12 0001C      BNEQ    2$
                            54        01  88 0001E      BISB2   #1, RESULT                              ; 1302
                                      3A  11 00021      BRB     8$
                        57  8F        50  91 00023 2$:  CMPB    R0, #87                               ; 1303
                                      05  12 00027      BNEQ    3$
                            54        02  88 00029      BISB2   #2, RESULT                              ; 1304
                                      2F  11 0002C      BRB     8$
                        45  8F        50  91 0002E 3$:  CMPB    R0, #69                               ; 1305
                                      06  13 00032      BEQL    4$
                        50  8F        50  91 00034      CMPB    R0, #80                               ; 1306
                                      05  12 00038      BNEQ    5$
                            54        04  88 0003A 4$:  BISB2   #4, RESULT                              ; 1307
                                      1E  11 0003D      BRB     8$
                        44  8F        50  91 0003F 5$:  CMPB    R0, #68                               ; 1308
                                      06  13 00043      BEQL    6$
                        4C  8F        50  91 00045      CMPB    R0, #76                               ; 1309
                                      05  12 00049      BNEQ    7$
                            54        08  88 0004B 6$:  BISB2   #8, RESULT                              ; 1310
                                      0D  11 0004E      BRB     8$
                                00000000G 8F  DD 00050 7$:  PUSHL   #CLI$_IVPROT                        ; 1311
                    00000000G   00        01  FB 00056      CALLS   #1, LIB$STOP
                B3      53          55  F2 0005D 8$:  AOBLSS  R5, INDEX, 1$                          ; 1299
                            50          54  D0 00061      MOVL    RESULT, R0                            ; 1314
                                        04  00064        RET                                         ; 1315

; Routine Size:  101 bytes,    Routine Base:  $CODE$ + 0A78
```

```
;  1327          1316  1 ROUTINE set_home (vbn, desc) =
;  1328          1317  1 !++
;  1329          1318  1 !
;  1330          1319  1 ! This routine reads a homeblock, modifies it, and writes it back to the
;  1331          1320  1 ! volume.
;  1332          1321  1 !
;  1333          1322  1 ! Inputs:
;  1334          1323  1 !       vbn - current vbn to read
;  1335          1324  1 !       ods1 - 0 => ODS2
;  1336          1325  1 !              1 => ODS1
;  1337          1326  1 !       desc - descriptor for the volume
;  1338          1327  1 !
;  1339          1328  1 !--
;  1340          1329  2 BEGIN
;  1341          1330  2
;  1342          1331  2
;  1343          1332  2 LOCAL
;  1344          1333  2     iosb : VECTOR[4,WORD],      ! I/O Status Block for $QIOW
;  1345          1334  2     status;                     ! General status return
;  1346          1335  2
;  1347          1336  2 !
;  1348          1337  2 ! Read the homeblock.
;  1349          1338  2 !
;  1350        P 1339  2 status = $QIOW (CHAN = .channel,
;  1351        P 1340  2                 FUNC = IO$_READVBLK,    ! Read virtual block
;  1352        P 1341  2                 IOSB = iosb,
;  1353        P 1342  2                 P1   = buffer,          ! Place it in 'buffer'
;  1354        P 1343  2                 P2   = 512,             ! Read 512 bytes
;  1355          1344  2                 P3   = .vbn);           ! Starting at this virutal block
;  1356          1345  2 IF .status THEN status = .iosb[0];
;  1357          1346  2 IF NOT .status
;  1358          1347  2 THEN
;  1359          1348  3     BEGIN
;  1360          1349  3     SIGNAL(set$_hbread,         ! Error reading homeblock
;  1361          1350  3            1,
;  1362          1351  3            .desc,              ! for this volume
;  1363          1352  3            .status);           ! for this reason
;  1364          1353  3     RETURN false;
;  1365          1354  2     END;
;  1366          1355  2
;  1367          1356  2 !
;  1368          1357  2 ! Change the ACCESSED (LRU) value, if requested
;  1369          1358  2 !
;  1370          1359  2 IF .flags[qual_access]
;  1371          1360  2 THEN
;  1372          1361  2     IF .ods1 THEN buffer[hm1$b_lru_lim] = .acc_value    ! For ODS1
;  1373          1362  2     ELSE buffer[hm2$b_lru_lim] = .acc_value;            ! For ODS2
;  1374          1363  2
;  1375          1364  2 !
;  1376          1365  2 ! If the DATA_CHECK qualifier is set, check to see if ODS1 or ODS2.  If ODS1,
;  1377          1366  2 ! tell the user that DATA_CHECK is illegal.  Otherwise, set the bits.
;  1378          1367  2 !
;  1379          1368  2 IF .flags[qual_data]
;  1380          1369  2 THEN IF .ods1                                    ! If ODS1,
;  1381          1370  2 THEN SIGNAL(set$_notods2,                       ! tell user no
;  1382          1371  2            1,
;  1383          1372  2            $DESCRIPTOR('DATA_CHECK'))
```

```
;  1384          1373  2 ELSE
;  1385          1374  2     BEGIN
;  1386          1375  3     IF .dflags[data_read] THEN buffer[hm2$v_readcheck] = 1;
;  1387          1376  3     IF .dflags[data_noread] THEN buffer[hm2$v_readcheck] = 0;
;  1388          1377  3     IF .dflags[data_write] THEN buffer[hm2$v_writcheck] = 1;
;  1389          1378  3     IF .dflags[data_nowrite] THEN buffer[hm2$v_writcheck] = 0;
;  1390          1379  2     END;
;  1391          1380  2
;  1392          1381  2 !
;  1393          1382  2 ! [NO]ERASE_ON_DELETE only works for ODS2.
;  1394          1383  2 !
;  1395          1384  2 IF .flags[qual_erase]
;  1396          1385  2 THEN IF .ods1
;  1397          1386  2 THEN SIGNAL(set$_notods2, 1, %ASCID 'ERASE_ON_DELETE')
;  1398          1387  2 ELSE buffer[hm2$v_erase] = .flags[qual_erase_val];
;  1399          1388  2
;  1400          1389  2 !
;  1401          1390  2 ! For the EXTENSION qualifier, if ODS1, the field is only a byte long, so
;  1402          1391  2 ! the greatest value is 255.  If the user specified a larger value, tell the
;  1403          1392  2 ! user and return.  Otherwise, make the change.
;  1404          1393  2 !
;  1405          1394  2 IF .flags[qual_exte]
;  1406          1395  2 THEN IF .ods1
;  1407          1396  2 THEN
;  1408          1397  3     BEGIN                                          ! Start of ODS1
;  1409          1398  3     IF .exte_value GTR 255
;  1410          1399  3     THEN
;  1411          1400  4         BEGIN
;  1412          1401  4         SIGNAL(set$_valerr);
;  1413          1402  4         RETURN false;
;  1414          1403  4         END
;  1415          1404  3     ELSE buffer[hm1$b_extend] = .exte_value;
;  1416          1405  3     END                                            ! End of ODS1
;  1417          1406  2 ELSE buffer[hm2$w_extend] = .exte_value;          ! Change ODS2 extend
;  1418          1407  2
;  1419          1408  2 !
;  1420          1409  2 ! For FILE_PROTECTION, the location is different, depending on which type of
;  1421          1410  2 ! disk we have.
;  1422          1411  2 !
;  1423          1412  2 ! Also, a word about the value in FPROT_VALUE.  The high word,
;  1424          1413  2 ! FPROT_VALUE<16,16>, contains a mask indicating which groups are to
;  1425          1414  2 ! be changed (SYSTEM,OWNER,GROUP,WORLD), while the low word,
;  1426          1415  2 ! FPROT_VALUE<0,16>, contains the complement of the new protection for each group.
;  1427          1416  2 ! Thus, if FPROT_VALUE<16,16> is zero, then nothing is to be changed and
;  1428          1417  2 ! there's no need to go thru the Boolean algebra.
;  1429          1418  2 !
;  1430          1419  2 IF .flags[qual_fprot] AND (.fprot_value<16,16> NEQ 0)
;  1431          1420  2 THEN
;  1432          1421  2     IF .ods1
;  1433          1422  2     THEN                                           ! For ODS1
;  1434          1423  3     buffer[hm1$w_fileprot] = (.buffer[hm1$w_fileprot] AND NOT.fprot_value<16,16>)
;  1435          1424  3                           OR (NOT.fprot_value<0,16> AND .fprot_value<16,16>)
;  1436          1425  2     ELSE                                           ! For ODS2
;  1437          1426  3     buffer[hm2$w_fileprot] = (.buffer[hm2$w_fileprot] AND NOT.fprot_value<16,16>)
;  1438          1427  3                           OR (NOT.fprot_value<0,16> AND .fprot_value<16,16>);
;  1439          1428  2
;  1440          1429  2 !
```

```
;  1441          1430  2 ! [NO]HIGHWATER only works for ODS2.
;  1442          1431  2 !
;  1443          1432  2 IF .flags[qual_fhw]
;  1444          1433  2 THEN IF .ods1
;  1445          1434  2 THEN SIGNAL(set$_notods2, 1, %ASCID 'HIGHWATER_MARKING')
;  1446          1435  2 ELSE buffer[hm2$v_nohighwater] = .flags[qual_fhw_val];
;  1447          1436  2
;  1448          1437  2 !
;  1449          1438  2 ! In the case of LABEL, the label is stored in the same place on both ODS1 and
;  1450          1439  2 ! ODS2 disks.  However, there is an additional field in ODS1 homeblocks, which
;  1451          1440  2 ! contain the volume label, padded with zeroes instead of blanks.
;  1452          1441  2 !
;  1453          1442  2 IF .flags[qual_label]
;  1454          1443  2 THEN
;  1455          1444  3     BEGIN
;  1456          1445  3     IF NOT .flags[qual_lbl_cpy]          ! If old label not copied
;  1457          1446  3     THEN
;  1458          1447  4         BEGIN                            ! then do so now.
;  1459          1448  4         CH$MOVE(vcb$s_volname,
;  1460          1449  4                 buffer[hm1$t_volname2],
;  1461          1450  4                 label_buff);
;  1462          1451  4         flags[qual_lbl_cpy] = 1;
;  1463          1452  3         END;
;  1464          1453  3     CH$COPY(.label_value[0],             ! Copy label into VOLNAME2,
;  1465          1454  3             .label_value[1],
;  1466          1455  3             ' ',                         ! padding with spaces.
;  1467          1456  3             vcb$s_volname,
;  1468          1457  3             buffer[hm1$t_volname2]);
;  1469          1458  3     IF .ods1
;  1470          1459  3     THEN CH$COPY(.label_value[0],        ! For ODS1, also copy to VOLNAME,
;  1471          1460  3                  .label_value[1],
;  1472          1461  3                  0,                      ! padding with zeroes
;  1473          1462  3                  vcb$s_volname,
;  1474          1463  3                  buffer[hm1$t_volname]);
;  1475          1464  2     END;
;  1476          1465  2
;  1477          1466  2 !
;  1478          1467  2 ! For OWNER_UIC, the ODS2 homeblock allows a full 16 bits for group, and
;  1479          1468  2 ! another 16 bits for member.  In the case of ODS1 disks, each of these fields
;  1480          1469  2 ! is only 8 bits long.  Also, if fold long UIC's into <377,377> for ODS1 disks.
;  1481          1470  2 !
;  1482          1471  2 IF .flags[qual_owner]
;  1483          1472  2 THEN
;  1484          1473  3     BEGIN
;  1485          1474  3     IF .ods1
;  1486          1475  3     THEN
;  1487          1476  4         BEGIN
;  1488          1477  4         IF .uic_value<8,8> NEQ 0
;  1489          1478  4         OR .uic_value<24,8> NEQ 0
;  1490          1479  4         THEN
;  1491          1480  5             BEGIN
;  1492          1481  5             uic_value<0,8> = -1;
;  1493          1482  5             uic_value<8,8> = 0;
;  1494          1483  5             uic_value<16,8> = -1;
;  1495          1484  5             uic_value<24,8> = 0;
;  1496          1485  4             END;
;  1497          1486  4         buffer[hm1$w_volowner] = (.uic_value<16,8> ^8) + .uic_value<0,8>;
```

```
;  1498          1487  4          END
;  1499          1488  3      ELSE buffer[hm2$l_volowner] = .uic_value;
;  1500          1489  2      END;
;  1501          1490  2
;  1502          1491  2  !
;  1503          1492  2  ! The retention period is something that only exists for ODS2 volumes.  If
;  1504          1493  2  ! this volume is not an ODS2 disk, then signal an error.  Otherwise, set the
;  1505          1494  2  ! default retention periods.
;  1506          1495  2  !
;  1507          1496  2
;  1508          1497  2  IF .flags[qual_retent]
;  1509          1498  2  THEN
;  1510          1499  3      BEGIN
;  1511          1500  3      IF .ods1                                ! If ODS1 disk
;  1512          1501  3      THEN SIGNAL(set$_notods2,               ! Signal an error
;  1513          1502  3                  1,
;  1514          1503  3                  $DESCRIPTOR('/RETENTION'))      ! Saying it can't be done
;  1515          1504  3      ELSE
;  1516          1505  4          BEGIN
;  1517          1506  4          CH$MOVE(8, retmin_value, buffer[hm2$q_retainmin]);
;  1518          1507  4          CH$MOVE(8, retmax_value, buffer[hm2$q_retainmax]);
;  1519          1508  3          END;
;  1520          1509  2      END;
;  1521          1510  2
;  1522          1511  2  !
;  1523          1512  2  ! PROTECTION, the volume protection, is also stored in two different places in
;  1524          1513  2  ! the home blocks.  See the discussion of the protection value for
;  1525          1514  2  ! FILE_PROTECTION, above.
;  1526          1515  2  !
;  1527          1516  3  IF .flags[qual_vprot] AND (.vprot_value<16,16> NEQ 0)
;  1528          1517  2  THEN
;  1529          1518  2      IF .ods1
;  1530          1519  2      THEN                                ! For ODS1
;  1531          1520  3      buffer[hm1$w_protect] = (.buffer[hm1$w_protect] AND NOT.vprot_value<16,16>)
;  1532          1521  3                             OR (NOT.vprot_value<0,16> AND .vprot_value<16,16>)
;  1533          1522  2      ELSE                                ! For ODS2
;  1534          1523  3      buffer[hm2$w_protect] = (.buffer[hm2$w_protect] AND NOT.vprot_value<16,16>)
;  1535          1524  3                             OR (NOT.vprot_value<0,16> AND .vprot_value<16,16>);
;  1536          1525  2
;  1537          1526  2
;  1538          1527  2  !
;  1539          1528  2  ! WINDOWS is also in two different places.
;  1540          1529  2  !
;  1541          1530  2  IF .flags[qual_windows]
;  1542          1531  2  THEN
;  1543          1532  3      BEGIN
;  1544          1533  3      IF .ods1 THEN buffer[hm1$b_window] = .window_value  ! For ODS1
;  1545          1534  3      ELSE buffer[hm2$b_window] = .window_value;          ! For ODS2
;  1546          1535  2      END;
;  1547          1536  2
;  1548          1537  2  !
;  1549          1538  2  ! The USER_NAME is in the same place for both types of home blocks.
;  1550          1539  2  !
;  1551          1540  2  IF .flags[qual_username]
;  1552          1541  2  THEN CH$COPY(.user_value[0],            ! Copy the username to the homeblock
;  1553          1542  2                ,user_value[1],
;  1554          1543  2                 ' ',                     ! padded with spaces
```

```
    2                 hm2$s_ownername,
    2                 buffer[hm2$t_ownername]);
    2
    2 !
    2 ! Recompute the checksums
    2 !
    2
    2 checksum2(buffer, $BYTEOFFSET(hm2$w_checksum1));
    2 checksum2(buffer, $BYTEOFFSET(hm2$w_checksum2));
    2
    2 !
    2 ! Write the modified homeblock back to the disk
    2 !
  P 2 status = $QIOW (CHAN = .channel,
  P 2             FUNC = IO$_WRITEVBLK,   ! Read Virtual Block
  P 2             IOSB = iosb,
  P 2             P1   = buffer,          ! From 'buffer'
  P 2             P2   = 512,             ! Write 512 bytes
    2             P3   = .vbn);           ! To this virtual block
    2 IF .status THEN status = .iosb[0];
    2 IF NOT .status                      ! If not able to read
    2 THEN
    3     BEGIN
    3     SIGNAL(set$_hbwrite,            ! Error writing a homeblock
    3            1,
    3            .desc,                   ! to this disk
    3            .status);                ! for this reason
    3     RETURN false;                   ! don't mod database
    3     END
    2 ELSE RETURN true;
    1 END;


                                                                  .PSECT   $PLIT$,NOWRT,NOEXE,2

                     4B 43 45 48 43 5F 41 54 41 44  0040C P.ADP:  .ASCII   \DATA_CHECK\                    ;
                                                    00416         .BLKB    2
                                          0000000A  00418 P.ADO:  .LONG    10                              ;
                                          00000000' 0041C         .ADDRESS P.ADP                           ;
45 54 45 4C 45 44 5F 4E 4F 5F 45 53 41 52 45        00420 P.ADR:  .ASCII   \ERASE_ON_DELETE\<0>            ;
                                                00  0042F                                                  ;
                                          010E000F  00430 P.ADQ:  .LONG    17694735                        ;
                                          00000000' 00434         .ADDRESS P.ADR                           ;
49 4B 52 41 4D 5F 52 45 54 41 57 48 47 49 48        00438 P.ADT:  .ASCII   \HIGHWATER_MARKING\<0><0><0>    ;
                               00 00 00 47 4E       00447                                                  ;
                                          010E0011  0044C P.ADS:  .LONG    17694737                        ;
                                          00000000' 00450         .ADDRESS P.ADT                           ;
                     4E 4F 49 54 4E 45 54 45 52 2F  00454 P.ADV:  .ASCII   \/RETENTION\                    ;
                                                    0045E         .BLKB    2
                                          0000000A  00460 P.ADU:  .LONG    10                              ;
                                          00000000' 00464         .ADDRESS P.ADV                           ;


                                                                  .PSECT   $CODE$,NOWRT,2
```

```
                                 OFFC 00000 SET_HOME:
                                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1316
                          5B 00000000'  EF  9E 00002        MOVAB   P.ADD, R11                            ;
                          5A 00000000G  00  9E 00009        MOVAB   UIC_VALUE, R10                        ;
                          59 00000000G  00  9E 00010        MOVAB   LIB$SIGNAL, R9                        ;
                          58 00000000'  EF  9E 00017        MOVAB   FPROT_VALUE, R8                       ;
                          57 00000000'  EF  9E 0001E        MOVAB   FLAGS, R7                             ;
                          5E            08  C2 00025        SUBL2   #8, SP                                ;
                                        7E  7C 00028        CLRQ    -(SP)                                 ; 1344
                                        7E  D4 0002A        CLRL    -(SP)                                 ;
                                    04  AC  DD 0002C        PUSHL   VBN                                   ;
                                7E  0200  8F  3C 0002F        MOVZWL  #512, -(SP)                           ;
                                    20  A7  9F 00034        PUSHAB  BUFFER                                ;
                                        7E  7C 00037        CLRQ    -(SP)                                 ;
                                    20  AE  9F 00039        PUSHAB  IOSB                                  ;
                                        31  DD 0003C        PUSHL   #49                                   ;
                                  0224  C7  DD 0003E        PUSHL   CHANNEL                               ;
                                        7E  D4 00042        CLRL    -(SP)                                 ;
                  00000000G     00        0C  FB 00044        CALLS   #12, SYS$QIOW                         ;
                                56        50  D0 0004B        MOVL    R0, STATUS                            ;
                                06        56  E9 0004E        BLBC    STATUS, 1$                            ; 1345
                                56        6E  3C 00051        MOVZWL  IOSB, STATUS                          ;
                                10        56  E8 00054        BLBS    STATUS, 2$                            ; 1346
                                          56  DD 00057 1$:      PUSHL   STATUS                                ; 1352
                                    08    AC  DD 00059        PUSHL   DESC                                  ; 1351
                                          01  DD 0005C        PUSHL   #1                                    ; 1349
                              00000000G   8F  DD 0005E        PUSHL   #SET$_HBREAD                          ;
                                    0255  31  00064        BRW     33$                                   ;
                              11            67  01  E1 00067 2$:      BBC     #1, FLAGS, 4$                 ; 1359
                                        07  0221  C7  E9 0006B        BLBC    OD$1, 3$                      ; 1361
                                  4E    A7    FC  A8  90 00070        MOVB    ACC_VALUE, BUFFER+46          ;
                                        05  11  00075        BRB     4$                                    ;
                                  65    A7    FC  A8  90 00077 3$:      MOVB    ACC_VALUE, BUFFER+69          ; 1362
                              38            67  02  E1 0007C 4$:      BBC     #2, FLAGS, 9$                 ; 1368
                                        0F  0221  C7  E9 00080        BLBC    OD$1, 5$                      ; 1369
                                            5B  DD 00085        PUSHL   R11                           ; 1372
                                            01  DD 00087        PUSHL   #1                            ; 1370
                                  00000000G     8F  DD 00089        PUSHL   #SET$_NOTODS2                 ;
                                            69  03  FB 0008F        CALLS   #3, LIB$SIGNAL                ;
                                            24  11  00092        BRB     9$                            ;
                              04        04  A7  01  E1 00094 5$:      BBC     #1, DFLAGS, 6$                ; 1375
                                        4A  A7  01  88 00099        BISB2   #1, BUFFER+42                 ;
                              04        04  A7  03  E1 0009D 6$:      BBC     #3, DFLAGS, 7$                ; 1376
                                        4A  A7  01  8A 000A2        BICB2   #1, BUFFER+42                 ;
                              04        04  A7  02  E1 000A6 7$:      BBC     #2, DFLAGS, 8$                ; 1377
                                        4A  A7  02  88 000AB        BISB2   #2, BUFFER+42                 ;
                              04        04  A7  04  E1 000AF 8$:      BBC     #4, DFLAGS, 9$                ; 1378
                                        4A  A7  02  8A 000B4        BICB2   #2, BUFFER+42                 ;
                              21        01  A7  04  E1 000B8 9$:      BBC     #4, FLAGS+1, 11$              ; 1384
                                            10  C7  E9 000BD        BLBC    OD$1, 10$                     ; 1385
                                      0221  AB  9F 000C2        PUSHAB  P.ADD                         ; 1386
                                            01  DD 000C5        PUSHL   #1                            ;
                                  00000000G     8F  DD 000C7        PUSHL   #SET$_NOTODS2                 ;
                                            69  03  FB 000CD        CALLS   #3, LIB$SIGNAL                ;
                                            0C  11  000D0        BRB     11$                           ;
                    50      01  A7        01  05  EF 000D2 10$:      EXTZV   #5, #1, FLAGS+1, R0           ; 1387
              4A    A7            01        02  50  F0 000D8        INSV    R0, #2, #1, BUFFER+42         ;
```

```
               2B             67            03 E1 000DE 11$:   BBC     #3, FLAGS, 14$                              ; 1394
                              52  00000000G 00 D0 000E2        MOVL    EXTE_VALUE, R2                              ; 1398
                              1B       0221 C7 E9 000E9        BLBC    ODS1, 13$                                   ; 1395
                    000000FF  8F            52 D1 000EE        CMPL    R2, #255                                    ; 1398
                                            0C 15 000F5        BLEQ    12$
                                  007711EA  8F DD 000F7        PUSHL   #7803370                                    ; 1401
                              69            01 FB 000FD        CALLS   #1, LIB$SIGNAL
                                       01C2 31 00100           BRW     35$                                         ; 1402
                          4D  A7            52 90 00103 12$:   MOVB    R2, BUFFER+45                               ; 1404
                                            04 11 00107        BRB     14$                                         ; 1395
                          66  A7            52 B0 00109 13$:   MOVW    R2, BUFFER+70                               ; 1406
               40             67            04 E1 0010D 14$:   BBC     #4, FLAGS, 16$                              ; 1419
                                        02  A8 B5 00111        TSTW    FPROT_VALUE+2
                                            3B 13 00114        BEQL    16$
                              1C       0221 C7 E9 00116        BLBC    ODS1, 15$                                   ; 1421
                              51         44 A7 3C 0011B        MOVZWL  BUFFER+36, R1                               ; 1423
                              50         02 A8 3C 0011F        MOVZWL  FPROT_VALUE+2, R0
                              51            50 CA 00123        BICL2   R0, R1
                              50         02 A8 3C 00126        MOVZWL  FPROT_VALUE+2, R0                           ; 1424
                              52            68 3C 0012A        MOVZWL  FPROT_VALUE, R2
                              50            52 CA 0012D        BICL2   R2, R0
            44  A7            50            51 A9 00130        BISW3   R1, R0, BUFFER+36
                                            1A 11 00135        BRB     16$                                         ; 1423
                              51         56 A7 3C 00137 15$:   MOVZWL  BUFFER+54, R1                               ; 1426
                              50         02 A8 3C 0013B        MOVZWL  FPROT_VALUE+2, R0
                              51            50 CA 0013F        BICL2   R0, R1
                              50         02 A8 3C 00142        MOVZWL  FPROT_VALUE+2, R0                           ; 1427
                              52            68 3C 00146        MOVZWL  FPROT_VALUE, R2
                              50            52 CA 00149        BICL2   R2, R0
            56  A7            50            51 A9 0014C        BISW3   R1, R0, BUFFER+54
                21        01  A7            06 E1 00151 16$:   BBC     #6, FLAGS+1, 18$                            ; 1432
                              10       0221 C7 E9 00156        BLBC    ODS1, 17$                                   ; 1433
                                         34 AB 9F 0015B        PUSHAB  P.ADS                                       ; 1434
                                            01 DD 0015E        PUSHL   #1
                                  00000000G 8F DD 00160        PUSHL   #SET$_NOTODS2
                              69            03 FB 00166        CALLS   #3, LIB$SIGNAL
                                            0C 11 00169        BRB     18$
        50      01  A7            01        07 EF 0016B 17$:   EXTZV   #7, #1, FLAGS+1, R0                         ; 1435
    4A  A7          01            03        50 F0 00171        INSV    R0, #3, #1, BUFFER+42
                    29            67        05 E1 00177 18$:   BBC     #5, FLAGS, 20$                              ; 1442
                    0C        02  A7        06 E0 0017B        BBS     #6, FLAGS+2, 19$                            ; 1445
            14      A7      01F8  C7        0C 28 00180        MOVC3   #12, BUFFER+472, LABEL_BUFF                 ; 1449
                              02  A7     40 8F 88 00187        BISB2   #64, FLAGS+2                                ; 1451
        0C          20        08  B8     04 A8 2C 0018C 19$:   MOVC5   LABEL_VALUE, @LABEL_VALUE+4, #32, #12, -    ; 1457
                                       01F8 C7    00193                BUFFER+472
                              09       0221 C7 E9 00196        BLBC    ODS1, 20$                                   ; 1458
        0C          00        08  B8     04 A8 2C 0019B        MOVC5   LABEL_VALUE, @LABEL_VALUE+4, #0, #12, -     ; 1463
                                         2E A7    001A2                BUFFER+14
                                            67 95 001A4 20$:   TSTB    FLAGS                                       ; 1471
                                            2C 18 001A6        BGEQ    24$
                              23       0221 C7 E9 001A8        BLBC    ODS1, 23$                                   ; 1474
                                         01 AA 95 001AD        TSTB    UIC_VALUE+1                                 ; 1477
                                            05 12 001B0        BNEQ    21$
                                         03 AA 95 001B2        TSTB    UIC_VALUE+3                                 ; 1478
                                            07 13 001B5        BEQL    22$
                              6A  00FF00FF  8F D0 001B7 21$:   MOVL    #16711935, UIC_VALUE                        ; 1481
                              50         02 AA 9A 001BE 22$:   MOVZBL  UIC_VALUE+2, R0                             ; 1486
```

```
                50              50              08  78  001C2          ASHL    #8, R0, R0                                      ;
                                51              6A  9A  001C6          MOVZBL  UIC_VALUE, R1                                   ;
            3E  A7              50              51  A1  001C9          ADDW3   R1, R0, BUFFER+30                               ;
                                                04  11  001CE          BRB     24$                                             ;  1474
                        4C      A7              6A  D0  001D0  23$:     MOVL    UIC_VALUE, BUFFER+44                            ;  1488
                        21      01      A7      E9  001D4  24$:     BLBC    FLAGS+1, 26$                                    ;  1497
                        10    0221      C7      E9  001D8          BLBC    ODS1, 25$                                       ;  1500
                              48        AB      9F  001DD          PUSHAB  P.ADU                                           ;  1503
                                        01      DD  001E0          PUSHL   #1                                              ;  1501
                      00000000G         8F      DD  001E2          PUSHL   #SET$_NOTODS2                                   ;
                              69                03  FB  001E8          CALLS   #3, LIB$SIGNAL                                  ;
                                                0C  11  001EB          BRB     26$                                             ;
            68  A7      10      A8              08  28  001ED  25$:     MOVC3   #8, RETMIN_VALUE, BUFFER+72                     ;  1506
            70  A7      18      A8              08  28  001F3          MOVC3   #8, RETMAX_VALUE, BUFFER+80                     ;  1507
                42      01      A7              02  E1  001F9  26$:     BBC     #2, FLAGS+1, 28$                                ;  1516
                                  0E    A8      B5  001FE          TSTW    VPROT_VALUE+2                                   ;
                                                3D  13  00201          BEQL    28$                                             ;
                                1D    0221      C7  E9  00203          BLBC    ODS1, 27$                                       ;  1518
                                51      40      A7  3C  00208          MOVZWL  BUFFER+32, R1                                   ;  1520
                                50      0E      A8  3C  0020C          MOVZWL  VPROT_VALUE+2, R0                               ;
                                51              50  CA  00210          BICL2   R0, R1                                          ;
                                50      0E      A8  3C  00213          MOVZWL  VPROT_VALUE+2, R0                               ;  1521
                                52      0C      A8  3C  00217          MOVZWL  VPROT_VALUE, R2                                 ;
                                50              52  CA  0021B          BICL2   R2, R0                                          ;
            40  A7              50              51  A9  0021E          BISW3   R1, R0, BUFFER+32                               ;
                                                1B  11  00223          BRB     28$                                             ;  1520
                                51      54      A7  3C  00225  27$:     MOVZWL  BUFFER+52, R1                                   ;  1523
                                50      0E      A8  3C  00229          MOVZWL  VPROT_VALUE+2, R0                               ;
                                51              50  CA  0022D          BICL2   R0, R1                                          ;
                                50      0E      A8  3C  00230          MOVZWL  VPROT_VALUE+2, R0                               ;  1524
                                52      0C      A8  3C  00234          MOVZWL  VPROT_VALUE, R2                                 ;
                                50              52  CA  00238          BICL2   R2, R0                                          ;
                54  A7              50              51  A9  00238          BISW3   R1, R0, BUFFER+52                               ;
                    11      01      A7              03  E1  00240  28$:     BBC     #3, FLAGS+1, 30$                                ;  1530
                                07    0221      C7  E9  00245          BLBC    ODS1, 29$                                       ;  1533
                                4C      A7      28  A8  90  0024A          MOVB    WINDOW_VALUE, BUFFER+44                         ;
                                                05  11  0024F          BRB     30$                                             ;
                                64      A7      28  A8  90  00251  29$:     MOVB    WINDOW_VALUE, BUFFER+68                         ;  1534
                        0A      01      A7              01  E1  00256  30$:     BBC     #1, FLAGS+1, 31$                                ;  1540
        0C              20      24      B8      20  A8  2C  0025B          MOVC5   USER_VALUE, @USER_VALUE+4, #32, #12, -          ;  1545
                                                0204  C7  00262                  BUFFER+484                                      ;
                                                3A  DD  00265  31$:     PUSHL   #58                                             ;  1551
                                                20  A7  9F  00267          PUSHAB  BUFFER                                          ;
                              00000000G         00      02  FB  0026A          CALLS   #2, CHECKSUM2                                   ;
                                                7E    01FE  8F  3C  00271          MOVZWL  #510, -(SP)                             ;  1552
                                                20      A7  9F  00276          PUSHAB  BUFFER                                          ;
                              00000000G         00          02  FB  00279          CALLS   #2, CHECKSUM2                           ;
                                                            7E  7C  00280          CLRQ    -(SP)                                   ;  1562
                                                            7E  D4  00282          CLRL    -(SP)                                   ;
                                                        04  AC  DD  00284          PUSHL   VBN                                     ;
                                                    7E    0200  8F  3C  00287          MOVZWL  #512, -(SP)                             ;
                                                        20  A7  9F  0028C          PUSHAB  BUFFER                                  ;
                                                            7E  7C  0028F          CLRQ    -(SP)                                   ;
                                                        20  AE  9F  00291          PUSHAB  IOSB                                    ;
                                                            30  DD  00294          PUSHL   #48                                     ;
                                                    0224    C7  DD  00296          PUSHL   CHANNEL                                 ;
                                                            7E  D4  0029A          CLRL    -(SP)                                   ;
```

```
                00000000G  00             0C  FB 0029C          CALLS   #12, SYS$QIOW                                    ;
                           56             50  D0 002A3          MOVL    R0, STATUS                                      ;
                           06             56  E9 002A6          BLBC    STATUS, 32$                                     ; 1563
                           56             6E  3C 002A9          MOVZWL  IOSB, STATUS                                    ;
                           12             56  E8 002AC          BLBS    STATUS, 34$                                     ; 1564
                                          56  DD 002AF 32$:      PUSHL   STATUS                                          ; 1570
                                    08    AC  DD 002B1          PUSHL   DESC                                            ; 1569
                                          01  DD 002B4          PUSHL   #1                                              ; 1567
                                00000000G 8F  DD 002B6          PUSHL   #SET$_HBWRITE                                   ;
                                69        04  FB 002BC 33$:      CALLS   #4, LIB$SIGNAL                                  ;
                                          04  11 002BF          BRB     35$                                             ; 1573
                                50        01  D0 002C1 34$:      MOVL    #1, R0                                          ;
                                          04 002C4              RET                                                     ;
                                          50  D4 002C5 35$:      CLRL    R0                                              ; 1574
                                          04 002C7              RET                                                     ;

; Routine Size:  712 bytes,    Routine Base:  $CODE$ + 0ADD
```

```
;  1587          1575  1 ROUTINE set_ucbvcb (ucb) : NOVALUE =
;  1588          1576  1 !++
;  1589          1577  1 !
;  1590          1578  1 ! This routine is called in kernel mode, to modify the fields in the UCB and
;  1591          1579  1 ! VCB which correspond to changes made in the homeblock.  The address of the
;  1592          1580  1 ! UCB is passed as the input argument.
;  1593          1581  1 !
;  1594          1582  1 !--
;  1595          1583  2 BEGIN
;  1596          1584  2
;  1597          1585  2 MAP ucb : REF $BBLOCK;                          ! Define the UCB
;  1598          1586  2
;  1599          1587  2 BIND
;  1600          1588  2     orb = .ucb[ucb$l_orb] : $BBLOCK,            ! Define the ORB
;  1601          1589  2     vcb = .ucb[ucb$l_vcb] : $BBLOCK,            ! Define the VCB
;  1602          1590  2     devchar = ucb[ucb$l_devchar] : $BBLOCK;     ! and devchar longword
;  1603          1591  2
;  1604          1592  2 !
;  1605          1593  2 ! Go thru the UCB and VCB, making the same changes to it that were made
;  1606          1594  2 ! to the homeblock.  Note that, if the LABEL qualifier is set, the volume
;  1607          1595  2 ! label is changed in the homeblock and in the VCB, but the logical name
;  1608          1596  2 ! (DISK$label) is NOT CHANGED.
;  1609          1597  2 !
;  1610          1598  2
;  1611          1599  2 IF .flags[qual_access] AND (.acc_inc NEQ 0)
;  1612          1600  2 THEN vcb[vcb$b_lru_lim] = .vcb[vcb$b_lru_lim] + .acc_inc;
;  1613          1601  2
;  1614          1602  2 IF (.flags[qual_data] AND (.buffer[hm2$b_struclev] NEQ 1))
;  1615          1603  2 THEN
;  1616          1604  3     BEGIN
;  1617          1605  3     IF .dflags[data_read] THEN devchar[dev$v_rck] = 1;
;  1618          1606  3     IF .dflags[data_noread] THEN devchar[dev$v_rck] = 0;
;  1619          1607  3     IF .dflags[data_write] THEN devchar[dev$v_wck] = 1;
;  1620          1608  3     IF .dflags[data_nowrite] THEN devchar[dev$v_wck] = 0;
;  1621          1609  3     END;
;  1622          1610  2
;  1623          1611  2 IF .flags[qual_erase]
;  1624          1612  2 AND NOT .ods1
;  1625          1613  2 THEN vcb[vcb$v_erase] = .flags[qual_erase_val];
;  1626          1614  2
;  1627          1615  2 IF .flags[qual_exte]
;  1628          1616  2 THEN vcb[vcb$w_extend] = .exte_value;
;  1629          1617  2
;  1630          1618  2 IF .flags[qual_fprot] AND (.fprot_value<16,16> NEQ 0)
;  1631          1619  2 THEN vcb[vcb$w_fileprot] = (.vcb[vcb$w_fileprot] AND NOT .fprot_value<16,16>)
;  1632          1620  2                          OR (NOT.fprot_value<0,16> AND .fprot_value<16,16>);
;  1633          1621  2
;  1634          1622  2 IF .flags[qual_fhw]
;  1635          1623  2 AND NOT .ods1
;  1636          1624  2 THEN vcb[vcb$v_nohighwater] = .flags[qual_fhw_val];
;  1637          1625  2
;  1638          1626  2 IF .flags[qual_label]
;  1639          1627  2 THEN CH$COPY(.label_value[0],
;  1640          1628  2                ,label_value[1],
;  1641          1629  2                ' ',
;  1642          1630  2                vcb$s_volname,
;  1643          1631  2                vcb[vcb$t_volname]);
```

```
;  1644          1632  2
;  1645          1633  2 IF .flags[qual_mntver]
;  1646          1634  2 THEN vcb[vcb$v_mountver] = .flags[qual_mntver_val];
;  1647          1635  2
;  1648          1636  2 IF .flags[qual_owner]
;  1649          1637  2 THEN orb[orb$l_owner] = .uic_value;
;  1650          1638  2
;  1651          1639  3 IF .flags[qual_retent] AND (NOT .ods1)
;  1652          1640  2 THEN
;  1653          1641  3     BEGIN
;  1654          1642  3     CH$MOVE(8, retmin_value, vcb[vcb$q_retainmin]);
;  1655          1643  3     CH$MOVE(8, retmax_value, vcb[vcb$q_retainmax]);
;  1656          1644  2     END;
;  1657          1645  2
;  1658          1646  2 IF .flags[qual_unl]
;  1659          1647  2 THEN ucb[ucb$v_unload] = .flags[qual_unl_val];
;  1660          1648  2
;  1661          1649  3 IF .flags[qual_vprot] AND (.vprot_value<16,16> NEQ 0)
;  1662          1650  3 THEN orb[orb$w_prot] = (.orb[orb$w_prot] AND NOT .vprot_value<16,16>)
;  1663          1651  2                       OR (NOT.vprot_value<0,16> AND .vprot_value<16,16>);
;  1664          1652  2 orb[orb$v_prot_16] = 1;                          ! SOGW protection word
;  1665          1653  2
;  1666          1654  2 IF .flags[qual_windows]
;  1667          1655  2 THEN vcb[vcb$b_window] = .window_value;
;  1668          1656  2
;  1669          1657  2 RETURN;
;  1670          1658  1 END;


                                07FC 00000 SET_UCBVCB:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                    ;  1575
                          5A 00000000'  EF  9E 00002           MOVAB   FPROT_VALUE, R10
                          59 00000000'  EF  9E 00009           MOVAB   FLAGS, R9
                             57    04   AC  D0 00010           MOVL    UCB, R7                         ;  1588
                             58    1C   A7  D0 00014           MOVL    28(R7), R8
                             56    34   A7  D0 00018           MOVL    52(R7), R6                      ;  1589
                       0C    69         01  E1 0001C           BBC     #1, FLAGS, 1$                   ;  1599
                                  0220   C9  95 00020           TSTB    ACC_INC
                                         06  13 00024           BEQL    1$
                    49    A6    0220   C9  80 00026           ADDB2   ACC_INC, 73(R6)                 ;  1600
                    2E    69          02  E1 0002C 1$:         BBC     #2, FLAGS, 5$                   ;  1602
                          01    2D    A9  91 00030           CMPB    BUFFER+13, #1
                                         28  13 00034           BEQL    5$
                    05    04    A9     01  E1 00036           BBC     #1, DFLAGS, 2$                  ;  1605
                          3B    A7     8F  88 0003B           BISB2   #64, 59(R7)
                    05    04    A9     03  E1 00040 2$:        BBC     #3, DFLAGS, 3$                  ;  1606
                          3B    A7     8F  8A 00045           BICB2   #64, 59(R7)
                    05    04    A9     02  E1 0004A 3$:        BBC     #2, DFLAGS, 4$                  ;  1607
                          3B    A7     8F  88 0004F           BISB2   #128, 59(R7)
                    05    04    A9     04  E1 00054 4$:        BBC     #4, DFLAGS, 5$                  ;  1608
                          3B    A7     8F  8A 00059           BICB2   #128, 59(R7)
                    11    01    A9     04  E1 0005E 5$:        BBC     #4, FLAGS+1, 6$                 ;  1611
                                   0221  C9  E8 00063           BLBS    ODS1, 6$                        ;  1612
50        01        A9          01     05  EF 00068           EXTZV   #5, #1, FLAGS+1, R0             ;  1613
```

```
                53   A6          01          03          50  F0 0006E          INSV    R0, #3, #1, 83(R6)                          ;
                            08      69          03  E1  00074 68:             BBC     #3, FLAGS, 7$                               ; 1615
                                 3E A6 00000000G  00  B0  00078             MOVW    EXTE_VALUE, 62(R6)                          ; 1616
                            1F      69          04  E1  00080 7$:            BBC     #4, FLAGS, 8$                               ; 1618
                                            02  AA  B5  00084             TSTW    FPROT_VALUE+2
                                                1A  13  00087             BEQL    8$
                                        51  4A  A6  3C  00089             MOVZWL  74(R6), R1                                  ; 1619
                                        50  02  AA  3C  0008D             MOVZWL  FPROT_VALUE+2, R0
                                        51      50  CA  00091             BICL2   R0, R1
                                        50  02  AA  3C  00094             MOVZWL  FPROT_VALUE+2, R0                           ; 1620
                                        52      6A  3C  00098             MOVZWL  FPROT_VALUE, R2
                                        50      52  CA  0009B             BICL2   R2, R0
                        4A  A6          50          51  A9  0009E         BISW3   R1, R0, 74(R6)
                                    11  01  A9      06  E1  000A3 8$:         BBC     #6, FLAGS+1, 9$                         ; 1622
                                        0C  0221    C9  E8  000A8             BLBS    ODS1, 9$                                ; 1623
                50      01  A9          01          07  EF  000AD         EXTZV   #7, #1, FLAGS+1, R0                         ; 1624
                53  A6          01      04          50  F0  000B3         INSV    R0, #4, #1, 83(R6)
                            09      69          05  E1  000B9 9$:         BBC     #5, FLAGS, 10$                              ; 1626
                    0C          20  08  BA      04  AA  2C  000BD         MOVC5   LABEL_VALUE, @LABEL_VALUE+4, #32, #12, -    ; 1631
                                            14  A6      000C4                     20(R6)
                                        0C  02  A9  E9  000C6 10$:        BLBC    FLAGS+2, 11$                                ; 1633
                50      02  A9          01          01  EF  000CA         EXTZV   #1, #1, FLAGS+2, R0                         ; 1634
                53  A6          01      02          50  F0  000D0         INSV    R0, #2, #1, 83(R6)
                                        69  95  000D6 11$:                TSTB    FLAGS                                       ; 1636
                                        07  18  000D8                     BGEQ    12$
                                        68 00000000G  00  D0  000DA         MOVL    UIC_VALUE, (R8)                             ; 1637
                                        11      01  A9  E9  000E1 12$:        BLBC    FLAGS+1, 13$                            ; 1639
                                        0C  0221    C9  E8  000E5             BLBS    ODS1, 13$
                        6C  A6      10  AA          08  28  000EA         MOVC3   #8, RETMIN_VALUE, 108(R6)                   ; 1642
                        74  A6      18  AA          08  28  000F0         MOVC3   #8, RETMAX_VALUE, 116(R6)                   ; 1643
                            0C      02  A9          02  E1  000F6 13$:        BBC     #2, FLAGS+2, 14$                        ; 1646
                50      02  A9          01          03  EF  000FB         EXTZV   #3, #1, FLAGS+2, R0                         ; 1647
            65  A7          01          04          50  F0  00101         INSV    R0, #4, #1, 101(R7)
                            20      01  A9          02  E1  00107 14$:        BBC     #2, FLAGS+1, 15$                        ; 1649
                                                0E  AA  B5  0010C             TSTW    VPROT_VALUE+2
                                                1B  13  0010F             BEQL    15$
                                        51  18  A8  3C  00111             MOVZWL  24(R8), R1                                  ; 1650
                                        50  0E  AA  3C  00115             MOVZWL  VPROT_VALUE+2, R0
                                        51      50  CA  00119             BICL2   R0, R1
                                        50  0E  AA  3C  0011C             MOVZWL  VPROT_VALUE+2, R0                           ; 1651
                                        52  0C  AA  3C  00120             MOVZWL  VPROT_VALUE, R2
                                        50      52  CA  00124             BICL2   R2, R0
                        18  A8          50          51  A9  00127         BISW3   R1, R0, 24(R8)
                                0B  A8              01  88  0012C 15$:        BISB2   #1, 11(R8)                              ; 1652
                            05      01  A9          03  E1  00130             BBC     #3, FLAGS+1, 16$                        ; 1654
                                    48  A6      28  AA  90  00135             MOVB    WINDOW_VALUE, 72(R6)                    ; 1655
                                                    04  0013A 16$:            RET                                             ; 1658

; Routine Size:  315 bytes,    Routine Base:  $CODE$ + 0DA5
```

```
;  1672          1659  1 ROUTINE modify_volset (desc) : NOVALUE =
;  1673          1660  2 BEGIN
;  1674          1661  2
;  1675          1662  2 !++
;  1676          1663  2 !
;  1677          1664  2 ! Modify [0,0]VOLSET.SYS on the root volume of the volume set.
;  1678          1665  2 ! Only ODS2 initialized volumes can be volume sets so we don't
;  1679          1666  2 ! have to worry about the $READ finding the End-of-File value
;  1680          1667  2 ! as zero in this case
;  1681          1668  2 !
;  1682          1669  2 ! Inputs:
;  1683          1670  2 !       desc - address of root volume device descriptor
;  1684          1671  2 !
;  1685          1672  2 ! Outputs:
;  1686          1673  2 !       None.
;  1687          1674  2 !
;  1688          1675  2 !--
;  1689          1676  2
;  1690          1677  2 MAP
;  1691          1678  2     desc : REF VECTOR;
;  1692          1679  2
;  1693          1680  2 LOCAL
;  1694          1681  2     status,
;  1695          1682  2     buffer : VECTOR[vsl$c_length,BYTE],
;  1696        P 1683  2     fab : $FAB(DNM = '[0,0]VOLSET.SYS',
;  1697          1684  2                FAC = <get,put,upd>),
;  1698        P 1685  2     rab : $RAB(FAB = fab,
;  1699        P 1686  2                UBF = buffer,
;  1700          1687  2                USZ = 100);
;  1701          1688  2
;  1702          1689  2 !
;  1703          1690  2 ! Put the root device name in place
;  1704          1691  2 !
;  1705          1692  2 fab[fab$l_fna] = .desc[1];
;  1706          1693  2 fab[fab$b_fns] = .desc[0];
;  1707          1694  2
;  1708          1695  2 !
;  1709          1696  2 ! Open and connect to [0,0]VOLSET.SYS
;  1710          1697  2 !
;  1711          1698  2 IF (status = $OPEN(FAB = fab))
;  1712          1699  2 THEN status = $CONNECT(RAB = rab);
;  1713          1700  2 IF NOT .status
;  1714          1701  2 THEN
;  1715          1702  2     BEGIN
;  1716          1703  2     LOCAL
;  1717          1704  2         ptr,
;  1718          1705  2         d : VECTOR[2],
;  1719          1706  2         b : VECTOR[30];
;  1720          1707  2     ptr = CH$MOVE(.fab[fab$b_fns],
;  1721          1708  2                   .fab[fab$l_fna],
;  1722          1709  2                   b);
;  1723          1710  2     ptr = CH$MOVE(.fab[fab$b_dns],
;  1724          1711  2                   .fab[fab$l_dna],
;  1725          1712  2                   .ptr);
;  1726          1713  2     SIGNAL(set$_writeerr,
;  1727          1714  2            1,
;  1728          1715  2            d,
```

```
;  1729          1716  3                .status);
;  1730          1717  3            END
;  1731          1718  3
;  1732          1719  2        ELSE
;  1733          1720  3            BEGIN
;  1734          1721  3
;  1735          1722  3        !
;  1736          1723  3        ! The first record contains the volume set name.  Skip it.
;  1737          1724  3        !
;  1738          1725  3            $GET(RAB = rab);
;  1739          1726  3
;  1740          1727  3        !
;  1741          1728  3        ! Search thru the records until the one matching the saved old label
;  1742          1729  3        ! is found.  When found, replace the old label with the new one, and
;  1743          1730  3        ! update the record.
;  1744          1731  3        !
;  1745          1732  3            WHILE $GET(RAB = rab) DO
;  1746          1733  4                BEGIN
;  1747          1734  4                IF CHSEQL(vcb$s_volname,
;  1748          1735  4                          label_buff,
;  1749          1736  4                          vsl$s_name,
;  1750          1737  4                          buffer,
;  1751          1738  4                          ' ')
;  1752          1739  4                THEN
;  1753          1740  5                    BEGIN
;  1754          1741  5                    CH$COPY(.label_value[0],
;  1755          1742  5                            ,label_value[1],
;  1756          1743  5                            ' ',
;  1757          1744  5                            vsl$s_name,
;  1758          1745  5                            buffer);
;  1759          1746  5                    rab[rab$l_rbf] = buffer;
;  1760          1747  5                    rab[rab$w_rsz] = vsl$c_length;
;  1761          1748  5                    $UPDATE(RAB = rab);
;  1762          1749  5                    EXITLOOP
;  1763          1750  4                    END;
;  1764          1751  3                END;
;  1765          1752  2            END;
;  1766          1753  2
;  1767          1754  2        $CLOSE(FAB = fab);
;  1768          1755  2
;  1769          1756  2        RETURN;
;  1770          1757  1        END;


                                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

53  59  53  2E  54  45  53  4C  4F  56  5D  30  2C  30  5B  00468 P.ADW:  .ASCII  \[0,0]VOLSET.SYS\                ;
                                                          00477         .BLKB   1                                 ;
                                                      03  00478 P.ADX:  .BYTE   3                                 ;
                                                      50  00479         .BYTE   80                                ;
                                                    0000  0047A         .WORD   0                                 ;
                                                00000000  0047C         .LONG   0                                 ;
                                                00000000  00480         .LONG   0                                 ;
                                                00000000  00484         .LONG   0                                 ;
                                                00000000  00488         .LONG   0                                 ;
                                                    0000  0048C         .WORD   0                                 ;
```

```
                    0048E           .BYTE   11
                    0048F           .BYTE   0
                00000490            .LONG   0
                    00494           .BYTE   0
                    00495           .BYTE   0
                    00496           .BYTE   0
                    00497           .BYTE   2
                00000498            .LONG   0
                0000049C            .LONG   0
                000004A0            .LONG   0
                000004A4            .LONG   0
                000004AB'           .ADDRESS P.ADW
                    004AC           .BYTE   0
                    004AD           .BYTE   15
                0000 004AE          .WORD   0
                00000000 004B0      .LONG   0
                0000 004B4          .WORD   0
                    004B6           .BYTE   0
                    004B7           .BYTE   0
                00000000 004B8      .LONG   0
                00000000 004BC      .LONG   0
                0000 004C0          .WORD   0
                    004C2           .BYTE   0
                    004C3           .BYTE   0
                00000000 004C4      .LONG   0
                    004C8 P.ADY:    .BYTE   1
                    004C9           .BYTE   68
                0000 004CA          .WORD   0
                00000000 004CC      .LONG   0
                00000000 004D0      .LONG   0
                00000000 004D4      .LONG   0
                0000# 004D8         .WORD   0[3]
                0000 004DE          .WORD   0
                00000000 004E0      .LONG   0
                0000 004E4          .WORD   0
                    004E6           .BYTE   0
                    004E7           .BYTE   0
                0064 004E8          .WORD   100
                0000 004EA          .WORD   0
                00000000 004EC      .LONG   0
                00000000 004F0      .LONG   0
                00000000 004F4      .LONG   0
                00000000 004F8      .LONG   0
                    004FC           .BYTE   0
                    004FD           .BYTE   0
                    004FE           .BYTE   0
                    004FF           .BYTE   0
                00000000 00500      .LONG   0
                00000000 00504      .LONG   0
                00000000 00508      .LONG   0

                                    .EXTRN  SYS$CONNECT, SYS$GET
                                    .EXTRN  SYS$UPDATE, SYS$CLOSE

                                    .PSECT  $CODE$,NOWRT,2

            00FC 00000 MODIFY_VOLSET:
```

```
                                57 00000000G   00    9E 00002                MOVAB   SYS$GET, R7                                     ;  1659
                                5E      FEAC   CE    9E 00009                MOVAB   -340(SP), SP                                   ;
                FF70    CD 00000000'   EF      0050   8F 28 0000E            MOVC3   #80, P.ADX, FAB                                 ;  1684
                0080    CE 00000000'   EF      0044   8F 28 0001A            MOVC3   #68, P.ADY, RAB                                 ;  1687
                            00A4       CE        C0   AD 9E 00026            MOVAB   BUFFER, RAB+36                                  ;  1684
                            FF68       CD      FF70   CD 9E 0002C            MOVAB   FAB, RAB+60                                     ;
                                       50        04   AC D0 00033            MOVL    DESC, R0                                        ;  1692
                                    9C   AD        04   A0 D0 00037            MOVL    4(R0), FAB+44                                   ;
                                    A4   AD              60 90 0003C            MOVB    (R0), FAB+52                                    ;  1693
                                                FF70   CD 9F 00040            PUSHAB  FAB                                             ;  1698
                            00000000G   00              01 FB 00044            CALLS   #1, SYS$OPEN                                    ;
                                       50              50 D0 0004B            MOVL    R0, STATUS                                      ;
                                       11              56 E9 0004E            BLBC    STATUS, 1$                                      ;
                                                  0080 CE 9F 00051            PUSHAB  RAB                                             ;  1699
                            00000000G   00              01 FB 00055            CALLS   #1, SYS$CONNECT                                 ;
                                       50              50 D0 0005C            MOVL    R0, STATUS                                      ;
                                       28              56 E8 0005F            BLBS    STATUS, 2$                                      ;  1700
                                       50        A4    AD 9A 00062 1$:         MOVZBL  FAB+52, R0                                      ;  1707
                        6E          9C   BD              50 28 00066            MOVC3   R0, @FAB+44, B                                  ;
                                       50        A5    AD 9A 0006B            MOVZBL  FAB+53, R0                                      ;  1710
                        63          A0   BD              50 28 0006F            MOVC3   R0, @FAB+48, (PTR)                              ;  1712
                                                        56 DD 00074            PUSHL   STATUS                                          ;  1716
                                                  7C    AE 9F 00076            PUSHAB  0                                               ;  1713
                                                        01 DD 00079            PUSHL   #1                                              ;
                                            00000000G   8F DD 0007B            PUSHL   #SET$_WRITEERR                                  ;
                            00000000G   00              04 FB 00081            CALLS   #4, LIB$SIGNAL                                  ;
                                                        42 11 00088            BRB     4$                                              ;  1700
                                                  0080  CE 9F 0008A 2$:         PUSHAB  RAB                                             ;  1725
                                                67      01 FB 0008E            CALLS   #1, SYS$GET                                     ;
                                                  0080  CE 9F 00091 3$:         PUSHAB  RAB                                             ;  1732
                                                67      01 FB 00095            CALLS   #1, SYS$GET                                     ;
                                                31      50 E9 00098            BLBC    R0, 4$                                          ;
                                CO      AD 00000000'   EF                      CMPC3   #12, LABEL_BUFF, BUFFER                         ;  1734
                                                       EB 12 000A4            BNEQ    3$                                              ;
                        0C              20 00000000'   FF 00000000'   EF   2C 000A6            MOVC5   LABEL_VALUE, @LABEL_VALUE+4, #32, #12, -        ;  1741
                                                                 CO    AD      000B3                    BUFFER                          ;
                                                00A8   CE        CO    AD 9E 000B5            MOVAB   BUFFER, RAB+40                  ;  1746
                                                00A2   CE        40    8F 9B 000BB            MOVZBW  #64, RAB+34                     ;  1747
                                                                 0080  CE 9F 000C1            PUSHAB  RAB                             ;  1748
                                            00000000G   00              01 FB 000C5            CALLS   #1, SYS$UPDATE                  ;
                                                                 FF70  CD 9F 000CC 4$:         PUSHAB  FAB                             ;  1754
                                            00000000G   00              01 FB 000D0            CALLS   #1, SYS$CLOSE                   ;
                                                                       04 000D7            RET                                     ;  1757

; Routine Size:  216 bytes,    Routine Base:  $CODE$ + 0EE0
```

```
;  1772          1758  1 GLOBAL ROUTINE COMMON_IO (EFN,CHAN,FUNC,IOSTS,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
;  1773          1759  1
;  1774          1760  1 !++
;  1775          1761  1 !
;  1776          1762  1 ! FUNCTIONAL DESCRIPTION:
;  1777          1763  1 !
;  1778          1764  1 !       This routine simply executes a $QIOW call with the parameters
;  1779          1765  1 !       supplied. It is called by the MOUNT code that SET links with.
;  1780          1766  1 !
;  1781          1767  1 ! CALLING SEQUENCE:
;  1782          1768  1 !       COMMON_IO (EFN,CHAN,FUNC,IOSTS,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)
;  1783          1769  1 !
;  1784          1770  1 ! INPUT PARAMETERS:
;  1785          1771  1 !       As to $QIOW
;  1786          1772  1 !
;  1787          1773  1 ! IMPLICIT INPUTS:
;  1788          1774  1 !       NONE
;  1789          1775  1 !
;  1790          1776  1 ! OUTPUT PARAMETERS:
;  1791          1777  1 !       NONE
;  1792          1778  1 !
;  1793          1779  1 ! IMPLICIT OUTPUTS:
;  1794          1780  1 !       NONE
;  1795          1781  1 !
;  1796          1782  1 ! ROUTINE VALUE:
;  1797          1783  1 !       As to $QIOW
;  1798          1784  1 !
;  1799          1785  1 ! SIDE EFFECTS:
;  1800          1786  1 !       As to $QIOW
;  1801          1787  1 !
;  1802          1788  1 !--
;  1803          1789  1
;  1804          1790  2 BEGIN
;  1805          1791  2
;  1806          1792  2 BUILTIN
;  1807          1793  2         AP,
;  1808          1794  2         CALLG;
;  1809          1795  2
;  1810          1796  2 EXTERNAL ROUTINE
;  1811          1797  2         SYS$QIOW        : ADDRESSING_MODE (GENERAL);
;  1812          1798  2
;  1813          1799  2
;  1814          1800  2 ! We simply pass the call and its parameters along to $QIOW.
;  1815          1801  2 !
;  1816          1802  2
;  1817          1803  2 CALLG (.AP, SYS$QIOW)
;  1818          1804  2
;  1819          1805  1 END;                                   ! End of routine COMMON_IO


                                        0000 00000      .ENTRY  COMMON_IO, Save nothing         ; 1758
                00000000G  00         6C  FA 00002      CALLG   (AP), SYS$QIOW                  ; 1803
                                        04 00009      RET                                     ; 1805
```

```
; Routine Size:  10 bytes,    Routine Base:  $CODE$ + 0FBB
```

```
; 1821          1806  1 END
; 1822          1807  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL, LIB$STOP

;                               PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $GLOBAL$                       48  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                         984  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                       1292  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                       4034  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                           Library Statistics
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619       181         0        1000       00:01.8
;  _$255$DUA28:[SYSLIB]TPAMAC.L32;1          42         0         0          14       00:00.2


;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:SETVOLUME/OBJ=OBJ$:SETVOLUME MSRC$:SETVOLUME/UPDATE=(ENH$:SETVOLUME)

; Size:          4034 code + 2324 data bytes
; Run Time:         01:09.3
; Elapsed Time:     03:48.1
; Lines/CPU Min:     1564
; Lexemes/CPU-Min: 23510
; Memory Used:  478 pages
; Compilation Complete
```

SETPROCES
LIS

SETSHOBRO
LIS

SETVOLUME
LIS

SETPWD
LIS

SETTERM
LIS

SETQUEUE
LIS

SETTIME
LIS

SHODEVPRT
LIS

SHODEVUTL
LIS

SHODEV
LIS

SHODEVCLU
LIS